

Réseaux informatiques

Nouvelle édition en français

Cisco CCNA

Guide de préparation à l'examen de
certification CCNA 200-301

Volume 1

**Fondamentaux
TCP/IP**

© 2021 François-Emmanuel Goffinet

Cisco CCNA 200-301 Volume 1

Guide de préparation au Cisco CCNA 200-301 en français, Volume 1
Fondamentaux TCP/IP

François-Emmanuel Goffinet

Ce livre est en vente à <http://leanpub.com/cisco-ccna-1>

Version publiée le 2021-06-16



Ce livre est publié par [Leanpub](#). Leanpub permet aux auteurs et aux éditeurs de bénéficier du Lean Publishing. [Lean Publishing](#) consiste à publier à l'aide d'outils très simples de nombreuses itérations d'un livre électronique en cours de rédaction, d'obtenir des retours et commentaires des lecteurs afin d'améliorer le livre.

© 2020 - 2021 François-Emmanuel Goffinet

Aussi par François-Emmanuel Goffinet

Cisco CCNA 200-301 Volume 2

Cisco CCNA 200-301 Volume 3

Cisco CCNA 200-301 Volume 4

Linux Administration Volume 1

Linux Administration Volume 2

Linux Administration Volume 3

Linux Administration Volume 4

Protocole SIP

Table des matières

Avertissement	i
Copyrights	i
Dédicace	ii
Remerciements	iii
Avant-Propos	iv
Cisco CCNA 200-301	v
Sujets et objectifs de l'examen Cisco CCNA 200-301	v
1.0 Fondamentaux des Réseaux - 20%	vi
2.0 Accès au Réseau - 20%	vii
3.0 Connectivité IP - 25%	vii
4.0 Services IP - 10%	viii
5.0 Sécurité de base - 15%	viii
6.0 Automation et Programmabilité - 10%	ix
Introduction	x
 Première partie Fondamentaux des réseaux	 1
1. Protocoles et modèles de communication	2
1. Définition d'un protocole de communication	2
2. Standardisation, régulation et interopérabilité	2
2.1. Organismes de standardisation	2
2.2. Organismes de régulation et consortiums commerciaux	2
3. Nomenclature des protocoles	3
3.1. Signalisation	3
3.2. Maintenance de la connexion	3
3.3. Fiabilité	4
3.4. Plans	4
4. Modèles de communication	4
4.1. Avantages d'un modèle de communication en couches	5
4.2. Modèles de communication utilisés dans ce document	5
4.3. Protocoles TCP/IP	5
4.4. Protocoles LAN et WAN	6
5. Modèles de conception	7
5.1. Modèles de conception traditionnels	8
5.2. Intent-Based Networks	8
6. Éléments clés à retenir	9
 2. Modèles TCP/IP et OSI	 10
1. Modèle TCP/IP	10
1.1. L'Internet	10

1.2. Objectifs de TCP/IP	12
1.3. Modèle TCP/IP	12
2. Modèle TCP/IP à quatre couches	13
2.1. Couche Application	13
2.2. Couche Transport : TCP	13
2.2. Couche Internet : IP	14
2.3. Couche Accès au réseau : LAN/WAN	14
2.4. Encapsulation	14
3. Illustration par un exemple	15
3.1. Couche Application	16
3.2. Couche Transport	17
3.3. Couche Internet	17
3.4. Socket TCP/IP	17
3.5. Couche Accès Réseau	18
3.6. Résumé des opérations	19
3.7. Désencapsulation du trafic à la réception	19
3.8. Communication d'égal à égal	20
4. Modèles et protocoles en détail	20
4.1. Modèle OSI et comparaison au modèle TCP/IP	21
4.2. Adressage, identifiants et matériel	21
4.3. Tableau de synthèse	21
5. Éléments clés à retenir	22
3. Composants de base du réseau	23
1. Rôles des périphériques	23
2. Les périphériques du réseau sont des ordinateurs	23
3. Composants d'un réseau associés aux couches OSI	24
4. Périphérique terminal	24
5. Routeur (<i>router</i>)	25
6. Commutateur d'entreprise (<i>switch</i>)	25
7. Commutateur multicouche (<i>Multilayers switch</i>)	26
8. Ponts, Concentrateurs et Répéteurs	26
9. Matériel sans-fil	27
10. Pare-feu	27
11. IPS	28
12. Supports de transmission	29
13. L'informatique en nuage	29
14. Cisco DNA	30
15. Éléments clés à retenir	31
4. Topologies du réseau	32
1. Topologies conceptuelles	32
1.1. Topologies simples en bus, point à point et en anneau	32
1.2. Topologie en étoile (star) et en étoile étendue	33
1.3. Topologie maillée (total mesh) et Topologie hybride (partial mesh)	33
2. Topologies de base	34
2.1. Un LAN d'entreprise	34
2.2. SOHO (Small Office / Home Office)	34
2.3. Deux sites interconnectés avec un Internet	35
2.4. Un Internet maillé	35
3. Topologies LAN	35
3.1. Modèle de conception Cisco Systems	35
3.2. Conception LAN avec VLANs	36
3.3. Topologie Campus LAN	36
4. Topologies WAN basiques	37
4.1. Connexion point à point sur une ligne dédiée entre deux sites	37

4.2. Un maillage WAN de trois routeurs	38
4.3. Topologie WAN tunnel GRE	38
5. Mathématiques des réseaux	39
1. Introduction	39
1.1. Objectifs	39
1.2. Systèmes de numération	39
1.3. Représentation	39
1.4. Unités de mesure	39
1.5. Codage des adresses réseau	40
2. Adresse IPv4	40
2.1. codage et étendue	40
2.2. Conversion décimale/binaire	40
2.3. Conversion binaire vers décimal	41
2.4. Valeurs possibles d'un octet dans un masque	41
2.5. Opération binaire ET	41
2.6. Exemple d'opérations binaires ET	41
2.7. Multiple	42
2.8. Logarithme binaire (en base 2)	42
2.9. Conversion d'un masque de réseau	42
3. Adresses MAC IEEE 802	42
3.1. Adresses MAC-48 IEEE 802	42
3.2. Adresses MAC EUI 64	43
4. Adresses IPv6	43
4.1. Blocs IPv6 /64	43
4.2. Blocs IPv6 /48	43
4.3. Conversion binaire / hexadécimal / decimal	44
5. Éléments clés à retenir	44
 Deuxième partie Cisco IOS CLI	 45
6. Cisco IOS Internetwork Operating System	46
1. Introduction	46
2. Versions	46
3. Trains	46
3.1. Jusqu'en version 12.4	46
3.2. Depuis 15.0	47
4. Packaging / feature sets	47
5. Cisco IOS	47
6. Cisco IOS-XR	48
7. Cisco IOS-XE	49
8. Cisco NX-OS	50
9. Cisco ASA OS	51
10. Cisco IOx	52
11. Sources du document	53
 7. Installer et configurer GNS3	 54
1. Le logiciel GNS3	54
1.1. Présentation de GNS3	54
1.2. Origine du projet	55
1.3. Considérations sur GNS3 et sur son usage	55
1.4. Alternatives	55
1.5. Fonctionnement	56
1.6. Versions de GNS3	56
2. Architecture Client / Serveur	56

2.1. Composants GNS3	56
2.2. Client GNS3 GUI	57
2.3. Serveur GNS3 Server	57
2.4. Types de déploiements du serveur GNS3	57
2.5. Installation de type Bare-Metal	57
2.6. Dimensionnement du serveur	57
2.7. Images pour GNS3	58
2.8. Périphériques intégrés	59
2.9. Exemples de topologies et de leur dimensionnement	60
2.10. Tableau comparatif des offres des fournisseurs IaaS	61
3. Installation de GNS3 Server chez Scaleway	61
3.1. Procédure d'installation de GNS3 SERVER	61
3.2. Inscription chez Scaleway	61
3.3. Créer une instance GP-BM1-S	62
3.4. Installation de GNS3 Server	62
3.5. Acquisition des images Cisco IOSv	62
3.6. Placement des images sur le serveur	63
3.7. Ultimes recommandations	63
3.8. Livre de jeu Ansible pour déployer GNS3 Server	63
4. Comment trouver des images Cisco IOS pour GNS3 ?	63
4.1. Abonnement Cisco VIRL	63
4.2. Images Cisco VIRL	63
4.3. Images Cisco pour les Labs	64
4.4. Alternative avec des images Dynamips	64
5. Installation de GNS3 GUI et connexion au serveur	64
5.1. Procédure d'installation de GNS3-GUI	64
5.2. Téléchargement et installation d'OpenVPN	65
5.3. Connexion OpenVPN	65
5.4. Téléchargement et installation de GNS3-GUI	65
5.5. Connexion à l'interface Web	67
6. Appliances GNS3	67
6.1. Définition d'une appliance GNS3	67
6.2. Marketplace GNS3	68
7. Création et manipulation d'un projet GNS3	68
7.1. Espace de travail (Workspace)	69
7.2. Barre d'outils (Toolbar)	70
7.3. Barre de périphériques (Devices Toolbar)	71
7.4. Panneau Résumé de la topologie (Topology Node Summary Pane)	72
7.5. Panneau Résumé des serveurs (Servers Summary Pane)	73
7.6. Panneau Console du contrôleur (Console Pane)	74
7.9. Connexion à la console	76
7.10. Option de gestion	76
8. Sujets avancés	77
8.1. Configuration de GNS3	77
8.2. Configuration des consoles	78
8.3. Capture du trafic et analyse Wireshark	78
8.4. Configurer les paramètres réseau d'un container Docker	78
8.5. Construction d'une "appliance" et d'une image personnalisée	78
8.6. Import / Export d'un projet "portable"	78
8.7. Connexion du PC local à la topologie distante	78
8.8. API HTTP REST de GNS3 Server	79
8.9. GNS3 as a Service	80
8. Connexion à un commutateur Cisco	82
1. Connexion série / console	82
2. Password Recovery	82

3. Password recovery sur un C2960	83
9. Connexion à un routeur Cisco	85
1. Connexion à un périphérique Cisco	85
1.1. Connexion série / console	85
1.2. Password Recovery	85
2. Password Recovery du routeur	85
2.1. Principe	86
2.2. Conditions	86
3. Procédure Password Recovery d'un routeur Cisco	86
3.1. Redémarrage du routeur et CTRL-PAUSE endéans les 60 secondes. Entrée dans le mode ROM Monitor	86
3.3. Redémarrage du routeur (et esquivé de la startup-configuration)	87
3.4. Visualisation de la configuration de démarrage	87
3.5. Vérification de la présence d'un fichier de configuration de démarrage	87
3.6. Modification des paramètres	88
3.7. Rétablissement du registre de démarrage et redémarrage	88
3.8. Vérification des paramètres du registre	88
4. Registre de configuration	88
4.1. Utilité	88
4.2. Signification des 16 bits du registre	88
4.3. Le bit 13 sur le premier hexa	89
4.4. Le bit 8 sur le deuxième hexa	89
4.5. Le bit 6 sur le troisième hexa	89
4.6. Les bits 3-2-1-0 sur le dernier hexa.	89
4.7. Les bits de vitesse (5-12-11)	90
4.8. Contrôle des adresses de Broadcast (bits 10 et 14)	90
10. Méthode Cisco IOS CLI	91
1. Introduction	91
2. Hiérarchie CLI	91
2.1. Passage en mode privilège	92
2.2. Passage en mode de configuration globale	93
2.3. Configuration d'une interface	93
2.4. Passage aux modes inférieurs	93
3. Aide, autocomplétion, historique, raccourcis, confort	93
3.1. Aide	93
3.2. Autocomplétion	94
3.3. Signalement d'erreurs	94
3.4. Commandes abrégées	94
3.5. Ambiguïté	95
3.6. Raccourcis clavier	95
4. Facilités à configurer	96
4.1. Commande do	96
4.2. Apparition des logs	96
4.3. Messages synchronisés de la console	96
4.4. Désactivation des recherches DNS	96
4.5. Filtrer les sorties des commandes show et more	97
 Troisième partie Protocole IPv4	 99
11. La couche Internet du modèle TCP/IP	100
1. Définition de la couche Internet	100
2. Modèle TCP/IP : couche Internet	100
3. Protocoles de couche Internet	101

4. Protocoles Internet	101
5. Internet : rôles	102
6. Routeur IP	102
7. Routage entre domaines IP	103
8. Organisation des adresses IP	103
9. Epuisement des adresses IPv4	104
10. Transition vers IPv6	106
11. Généalogie IPv4/IPv6	107
12. Nouveautés IPv6	108
12. En-têtes IPv4 et IPv6	109
1. En-tête IPv4	109
2. En-tête IPv6 de base	110
3. En-têtes IPv6 d'extension	111
4. Découverte du MTU dans le chemin	111
13. Introduction aux adresses IP	112
1. Adressage IP	112
2. Type d'adresses IP	112
3. Internet : adressage IPv4 et IPv6	113
4. Masque d'adresse	113
5. Partie réseau ou préfixe	114
6. Partie hôte ou identifiant d'interface	114
7. Première et dernière adresse IPv4	114
8. Adressage IPv6	114
9. Tables et protocoles	115
10. Passerelle par défaut	115
11. Adressage privé et adressage public en IPv4	115
12. Nécessité du NAT en IPv4	115
13. NAT et pare-feu	116
14. Gestion des adresses IP	116
14. Adressage IPv4	117
1. Introduction aux adresses IPv4	117
2. Définition	117
3. Identification de la classe d'adresse	117
Les Classes	117
Distinction de la partie réseau de la partie hôte	118
4. Utilisation d'un masque	118
Masque par défaut	119
5. Méthode par calcul binaire	119
Masques restrictifs	120
6. Méthode dite du nombre magique.	120
Nombre de sous-réseaux / nombre d'hôtes	121
7. Le routage sans classe (CIDR)	121
8. Notation CIDR	122
9. Masques à longueurs variables (VLSM)	122
10. Super-réseaux	124
11. Agrégation de routes	124
12. Protocoles de routage sans classe (Classless)	125
13. Exercices IPv4	125
Outil pour vérifier ses calculs	125
15. Protocoles ARP et ICMP	127
1. Introduction	127
1.1. ARP, ICMP, ICMPv6	127

1.2. Protocoles de résolution d'adresses et de découverte du voisinage	127
2. Résolution d'adresse ARP	127
2.1. Address Resolution Protocol	128
2.2. Résolution d'adresses en IPv4	128
2.3. Variantes ARP	128
2.4. Processus ARP	129
2.5. Empoisonnement de cache ARP	129
3. ICMP	129
3.1. Liste de messages ICMP	129
3.2. ICMP Types 8 Echo Request - 0 Echo Reply	130
3.3. ICMP Type 3 Destination Unreachable	130
3.4. Type 11 Time exceeded	131
3.5. Conclusions	131
16. Couche Transport TCP et UDP	132
1. Introduction	132
1.1. TCP	132
1.2. UDP	133
1.3. Comparaison UDP et TCP	133
1.4. En-tête TCP	134
1.5. En-tête UDP	134
2. Numéros de ports	134
3. Protocole TCP	135
3.1. TCP : fonctionnement	135
3.2. Machine à état TCP	136
3.3. TCP Three Way Handshake	136
3.4. Transfert des données	137
3.5. Numéro de séquence et numéro d'acquittement	138
3.6. Temporisation	138
3.7. Somme de contrôle	138
3.8. Contrôle de flux	138
3.9. Contrôle de congestion	138
3.10. Autres fonctionnalités TCP	139
3.11. Fin d'une connexion	139
4. Exemple de trafic TCP et UDP	139
5. Sources	140
17. Lab vérifications et analyses TCP/IP	141
1. Objectifs	141
1.1. Préalable	141
2. Construire la topologie	141
2.1. Composants	141
2.2. Connexions	141
2.3. Plan d'adressage	142
2.4. Configuration du routeur	142
3. Vérifications à partir des clients TCP/IP	143
3.1. Paramètres IPv4	143
4. Configuration manuelle d'un hôte terminal	146
5. Analyse de trafic	146
5.1. Trafic HTTP et DNS	146
5.2. Trafic TCP / UDP	147
Quatrième partie Adressage IPv6	148
18. Introduction aux adresses IPv6	149

1. Terminologie IPv6	149
2. Définition et étendue d'une adresse IPv6	149
3. Ecriture résumée	150
4. Types d'adresse IPv6	151
Adresses Unicast	152
Adresses Multicast	152
5. Méthodes de configuration des interfaces	152
6. Autoconfiguration Automatique	153
7. Autoconfiguration des identifiants d'interface	153
Identifiant d'interface MAC-EUI64 "Modified"	153
Privacy Extensions for Stateless Address Autoconfiguration in IPv6	155
8. Résumé	155
19. Prise d'information IPv6	156
1. Topologie	156
1.1. Présentation	156
1.2. Objectifs de l'activité	156
2. Hôte Windows	157
2.1. Liste des commandes Windows	157
2.2. ipconfig /all	157
2.3. netsh interface ipv6 show interface	159
2.4. netsh interface ipv6 show address	159
2.5. netsh interface ipv6 show address interface=X	160
2.6. netsh interface ipv6 show joins	161
2.7. netsh interface ipv6 show neighbor	162
2.8. netsh interface ipv6 show route	164
2.9. Get-NetIPInterface -AddressFamily IPv6	165
2.10. Get-NetIPAddress -AddressFamily IPv6	165
2.11. Get-NetNeighbor -AddressFamily IPv6	169
2.12. Get-NetRoute -AddressFamily IPv6	170
3. Hôte Linux	172
3.1. Liste des commandes	172
3.2. ip -6 add	172
3.3. ip -6 maddress	172
3.4. ip -6 neigh	173
3.5. ip -6 route	173
20. Les adresses IPv6 Unicast	174
1. Adresse IPv6 Unicast de bouclage (::1/128)	174
2. Adressage IPv6 Link-Local Unicast	175
2.1. Adresses IPv6 Link-Local (fe80::/10)	175
2.2. Caractéristiques d'une adresse IPv6 Link-Local	175
2.3. Reconnaître une adresse IPv6 Link-local	175
2.4. Utilité d'une adresse IPv6 Link-Local	176
2.5. Portée d'une adresse IPv6 Link-Local	176
3. Adressage IPv6 Global Unicast (2000::/3)	177
3.1. Adresses IPv6 Global Unicast	177
3.2. Format des adresses Global Unicast	177
3.3. Caractéristiques des adresses IPv6 Global Unicast	177
3.4. Attribution des adresses IPv6 Global Unicast	177
3.5. Comment reconnaître les adresses IPv6 Global Unicast ?	180
4. Adressage IPv6 Unique Local (FC00::/7)	181
4.1. Adresses IPv6 Unique Local	181
4.2. Caractéristiques des adresses IPv6 Unique Local	181
4.3. Utilité des adresses IPv6 Unique Local	181
4.4. Format des adresses IPv6 Unique Local	181

4.5. Comment reconnaître des adresses IPv6 Unique Local ?	182
4.6. Développement du RFC 4193	182
21. Les adresses IPv6 Multicast	183
1. Introduction aux adresses IPv6 Multicast FF00::/8	183
2. Types d'adresse Multicast	183
3. Format d'une adresse IPv6 Multicast	184
4. Identifiant de groupe	184
5. Identifier les groupes Multicast d'une interface IPv6	184
6. Adresse IPv6 Solicited-Node Multicast FF02::1:ff00:0/104	185
22. Configuration des interfaces IPv6 en Cisco IOS	186
1. Comment activer IPv6 sous Cisco IOS ?	186
2. Comment vérifier la connectivité locale IPv6 sous Cisco IOS ?	187
3. Comment consulter la table de voisinage ?	188
4. Comment configurer une adresse Link-Local statique ?	188
5. Questions bonus sur IPv6	189
Questions pratiques	189
Questions théoriques	189
23. Plans d'adressage IPv6	190
1. Introduction aux plans d'adresses IPv6	190
2. Avantages d'un plan d'adressage	190
3. Buts d'un plan d'adressage	190
4. Attribution des blocs d'adresses IPv6	190
5. Adresses IPv6 Provider-Aggregatable (PA) et Provider-Independent (PI)	191
6. Principe de découpage	191
7. Plan d'adressage Simple	191
8. Plan d'adressage à 2 ou 4 niveaux égaux	191
9. Plan d'adressage à plusieurs niveaux	192
10. Plan d'adressage en niveaux inégaux	193
11. Attribution étalée	193
12. Considérations pratiques sur les adresses IPv6	193
Révisions	195

Avertissement

Le projet lié à cet ouvrage est conçu principalement pour des candidats francophones à l'examen de certification Cisco CCNA 200-301.

Le document sera probablement utile comme *support de formation* dans d'autres contextes tels que celui de l'autoapprentissage, de l'enseignement ou de la formation professionnelle.

Si le document peut sans doute contribuer à mieux connaître les réseaux d'entreprise dans la perspective du CCNA, il ne peut aucunement garantir la réussite de l'examen. Aussi, ce projet n'a jamais poursuivi l'ambition de remplacer d'autres sources d'information/formation issues des canaux officiels tels que *Cisco Press*, *Cisco Learning Network*, les *Cisco Systems Learning Partners*, *Cisco Academy* ou encore la documentation officielle du fabricant. D'ailleurs l'auteur est totalement indépendant de tout fabricant cité. Celles-ci, toutes mieux présentées les unes que les autres, ne manquent pas au contraire, mais il est rare de trouver des sources de qualité et fiables en français.

Copyrights

Les entreprises suivantes et leurs marques protégées sont citées dans le document :

- Cisco Systems
- HP/Aruba
- VMWare
- Microsoft
- Red Hat
- Canonical
- Linux Foundation
- Wikimedia
- Wikipedia
- Docker
- GNS3

Dédicace

À mes parents qui m'ont toujours apporté un soutien sans faille dans tous mes projets.

Remerciements

Merci aux milliers de visiteurs quotidiens du site cisco.goffinet.org.

Merci aux centres de formation et aux écoles qui m'accordent leur confiance et qui me permettent de rencontrer mon public en personne.

Merci à [Wendell Odom](#), mon mentor sur le sujet Cisco CCNA. N'hésitez pas à vous procurer ses livres en anglais chez [Cisco Press](#).

Merci à [Stéphane Bortzmeyer](#) dont la prose prolifique m'inspire et m'aide à vulgariser les technologies de l'Internet.

Merci enfin à Cisco Systems d'être aussi ouvert depuis tant d'années dans sa documentation et pour son effort à rendre les technologies des réseaux plus accessibles, mieux comprises et plus populaires.

Avant-Propos

François-Emmanuel Goffinet est formateur IT et enseignant depuis 2002 en Belgique et en France. Outre Cisco CCNA, il couvre de nombreux domaines des infrastructures informatiques, du réseau à la virtualisation des systèmes, du nuage à la programmation d'infrastructures hétérogènes en ce y compris DevOps, Docker, K8s, chez AWS, GCP ou Azure, etc. avec une forte préférence et un profond respect pour l'Open Source, notamment pour Linux.

On trouvera ici un des résultats d'un projet d'autopublication en mode *agile* plus large lié au site web cis-co.goffinet.org. La documentation devrait évoluer dans un format vidéo. Les sujets développés devraient trouver des questionnaires de validation de connaissances. Enfin, une solution accessible et abordable de simulation d'exercices pratiques mériterait réflexion.

Cisco CCNA 200-301

L'examen [Cisco CCNA 200-301](#)¹ est disponible en anglais uniquement. Il se déroule sous surveillance dans un centre de test VUE après une inscription sur leur site [vue.com](#) et un paiement (de maximum 300 EUR) avec un bon de réduction (*voucher*) ou par carte de crédit.

Cet examen de niveau fondamental sur la théorie des réseaux évalue votre niveau avec un examen sur ordinateur en anglais constitué d'une centaine de questions théoriques et pratiques. Cet examen a une durée de 120 minutes. Il est interdit de revenir sur une question à laquelle on a déjà répondu. Le seuil de réussite est fixé entre 82,5% et 85%. Tout diplômé d'un premier cycle de l'enseignement supérieur en informatique devrait être en mesure de réussir cet examen dans un délai de trois mois. Tout qui voudrait entrer dans une carrière dans les réseaux ne perd pas son temps en passant cet examen. Certains prétendent même que c'est fortement recommandé.

Sujets et objectifs de l'examen Cisco CCNA 200-301

On trouve 53 objectifs dans six sujets² : Fondamentaux des Réseaux (20%), Accès au Réseau (20%), Connectivité IP (25%), Services IP (10%), Sécurité de base (15%), Automation et Programmabilité (10%).

On trouve aussi dix verbes dans les objectifs de la certification CCNA 200-301 qui correspondent à certaines compétences à valider :

1. "Expliquer" (6)
2. "Décrire" (15)
3. "Comparer" (6)
4. "Identifier" (1)
5. "Reconnaître" (1)
6. "Interpréter" (2)
7. "Déterminer" (1)
8. "Définir" (1)
9. "Configurer" (17)
10. "Vérifier" (1)

On peut considérer que seuls les objectifs qui demandent à "Configurer" et à "Vérifier" seraient purement pratiques. Toutefois, "Identifier", "Interpréter" et "Déterminer" pourraient aussi trouver leur application opérationnelle. Les autres objectifs comme "Expliquer", "Décrire", "Définir", "Reconnaître" seraient validés par des questions d'examen plus théoriques.

Les objectifs développés dans ce volume sont indiqués en gras.

1. La page officielle de la certification se trouve [à cette adresse](#).

2. La page officielle des sujets et des objectifs du Cisco CCNA 200-301 se trouve [à cette adresse](#).

1.0 Fondamentaux des Réseaux - 20%

- 1.1 Expliquer le rôle et la fonction des composants réseau
 - 1.1.a Routeurs
 - 1.1.b Commutateurs (switches) L2 et L3
 - 1.1.c Pare-feu NG (Next-generation firewalls) et IPS
 - 1.1.d Point d'accès (Access points)
 - 1.1.e Contrôleurs (Cisco DNA Center et WLC)
 - 1.1.f Points terminaux (Endpoints)
 - 1.1.g Serveurs
- 1.2 Décrire les caractéristiques des architectures et topologies réseau
 - 1.2.a 2 tier
 - 1.2.b 3 tier
 - 1.2.c Spine-leaf
 - 1.2.d WAN
 - 1.2.e Small office/home office (SOHO)
 - 1.2.f On-premises et cloud
- 1.3 Comparer les interfaces physiques et les types de câble
 - 1.3.a Fibre monmode (Single-mode) et fibre multimode, cuivre
 - 1.3.b Connexions (Ethernet shared media et point-to-point)
 - 1.3.c Concepts sur PoE
- 1.4 Identifier les problèmes d'interface et de câbles (collisions, errors, mismatch duplex, et/ou speed)
- 1.5 Comparer TCP à UDP
- 1.6 Configurer et vérifier l'adressage et le sous-réseauage (subnetting) IPv4
- 1.7 Décrire la nécessité d'un adressage IPv4 privé
- 1.8 Configurer et vérifier l'adressage et les préfixes IPv6
- 1.9 Comparer les types d'adresses IPv6
 - 1.9.a Global unicast
 - 1.9.b Unique local
 - 1.9.c Link local
 - 1.9.d Anycast
 - 1.9.e Multicast
 - 1.9.f Modified EUI 64
- 1.10 Vérifier les paramètres IP des OS clients (Windows, Mac OS, Linux)
- 1.11 Décrire les principes des réseaux sans-fil
 - 1.11.a Nonoverlapping Wi-Fi channels
 - 1.11.b SSID
 - 1.11.c RF
 - 1.11.d Encryption
- 1.12 Expliquer les fondamentaux de la virtualisation (virtual machines)
- 1.13 Décrire les concepts de la commutation (switching)
 - 1.13.a MAC learning et aging
 - 1.13.b Frame switching
 - 1.13.c Frame flooding
 - 1.13.d MAC address table

2.0 Accès au Réseau - 20%

- 2.1 Configurer et vérifier les VLANs (normal range) couvrant plusieurs switches
 - 2.1.a Access ports (data et voice)
 - 2.1.b Default VLAN
 - 2.1.c Connectivity
- 2.2 Configurer et vérifier la connectivité interswitch
 - 2.2.a Trunk ports
 - 2.2.b 802.1Q
 - 2.2.c Native VLAN
- 2.3 Configurer et vérifier les protocoles de découverte Layer 2 (Cisco Discovery Protocol et LLDP)
- 2.4 Configurer et vérifier (Layer 2/Layer 3) EtherChannel (LACP)
- 2.5 Décrire la nécessité et les opérations de base de Rapid PVST+ Spanning Tree Protocol
 - 2.5.a Root port, root bridge (primary/secondary), et les autres noms de port
 - 2.5.b Port states (forwarding/blocking)
 - 2.5.c Avantages PortFast
- 2.6 Comparer les architectures Cisco Wireless Architectures et les modes des APs
- 2.7 Décrire les connexions physiques d'infrastructure des composants WLAN (AP,WLC, access/trunk ports, et LAG)
- 2.8 Décrire les connexions des accès de gestion des APs et du WLC (Telnet, SSH, HTTP,HTTPS, console, et TACACS+/RADIUS)
- 2.9 Configurer les composants d'un accès au LAN sans-fil pour la connectivité d'un client en utilisant un GUI seulement pour la création du WLAN, les paramètres de sécurité, les profils QoS et des paramètres WLAN avancés

3.0 Connectivité IP - 25%

- 3.1 Interpréter les composants d'une table de routage
 - 3.1.a Routing protocol code
 - 3.1.b Prefix
 - 3.1.c Network mask
 - 3.1.d Next hop
 - 3.1.e Administrative distance
 - 3.1.f Metric
 - 3.1.g Gateway of last resort
- 3.2 Déterminer comment un routeur prend une décision de transfert par défaut
 - 3.2.a Longest match
 - 3.2.b Administrative distance
 - 3.2.c Routing protocol metric
- 3.3 Configurer et vérifier le routage statique IPv4 et IPv6
 - 3.3.a Default route
 - 3.3.b Network route

- 3.3.c Host route
 - 3.3.d Floating static
- 3.4 Configurer et vérifier single area OSPFv2
 - 3.4.a Neighbor adjacencies
 - 3.4.b Point-to-point
 - 3.4.c Broadcast (DR/BDR selection)
 - 3.4.d Router ID
- 3.5 Décrire le but des protocoles de redondance du premier saut (first hop redundancy protocol)

4.0 Services IP - 10%

- 4.1 Configurer et vérifier inside source NAT (static et pools)
- 4.2 Configurer et vérifier NTP dans le mode client et le mode server
- 4.3 Expliquer le rôle de DHCP et de DNS au sein du réseau
- 4.4 Expliquer la fonction de SNMP dans les opérations réseau
- 4.5 Décrire l'utilisation des fonctionnalités de syslog features en ce inclus les facilities et niveaux
- 4.6 Configurer et vérifier DHCP client et relay
- 4.7 Expliquer le forwarding per-hop behavior (PHB) pour QoS comme classification, marking, queuing, congestion, policing, shaping
- 4.8 Configurer les périphériques pour un accès distant avec SSH
- 4.9 Décrire les capacités la fonction de TFTP/FTP dans un réseau

5.0 Sécurité de base - 15%

- 5.1 Définir les concepts clé de la sécurité (menaces, vulnérabilités, exploits, et les techniques d'atténuation)
- 5.2 Décrire les éléments des programmes de sécurité (sensibilisation des utilisateurs, formation, le contrôle d'accès physique)
- 5.3 Configurer l'accès aux périphériques avec des mots de passe
- 5.4 Décrire les éléments des politiques de sécurité comme la gestion, la complexité, et les alternatives aux mots de passe (authentications multifacteur, par certificats, et biométriques)
- 5.5 Décrire les VPNs remote access et site-to-site
- 5.6 Configurer et vérifier les access control lists
- 5.7 Configurer les fonctionnalités de sécurité Layer 2 (DHCP snooping, dynamic ARP inspection, et port security)
- 5.8 Distinguer les concepts authentication, authorization, et accounting
- 5.9 Décrire les protocoles de sécurité sans-fil (WPA, WPA2, et WPA3)
- 5.10 Configurer un WLAN en utilisant WPA2 PSK avec un GUI

6.0 Automation et Programmabilité - 10%

- 6.1 Expliquer comment l'automation impacte la gestion du réseau
- 6.2 Comparer les réseaux traditionnels avec le réseau basé contrôleur (controller-based)
- 6.3 Décrire les architectures basées contrôleur (controller-based) et software defined (overlay, underlay, et fabric)
 - 6.3.a Séparation du control plane et du data plane
 - 6.3.b APIs North-bound et south-bound
- 6.4 Comparer la gestion traditionnelle des périphériques campus avec une gestion des périphériques avec Cisco DNA Center
- 6.5 Décrire les caractéristiques des APIs de type REST (CRUD, verbes HTTP, et encodage des données)
- 6.6 Reconnaître les capacités des mécanismes de gestion des configurations comme Puppet, Chef, et Ansible
- 6.7 Interpréter des données encodées en JSON

Introduction

Ce premier volume du guide de préparation à la certification Cisco CCNA 200-301 est une première étape dans votre projet de formation. Il couvre les objectifs de la certification qui demeurent indispensables à la suite. Très théoriques, ces objectifs trouvent de nombreux cas d'application dans des exercices de lab.

L'objectif opérationnel de ce document est d'identifier les composants des topologies du réseau et de maîtriser l'adressage IPv4 et l'adressage IPv6, aussi bien en termes de calculs de masques et de sous-réseau. On aura également un aperçu des commandes et des procédures de diagnostic de la connectivité TCP/IP. L'ouvrage s'intéresse au célèbre IOS Cisco et au simulateur d'infrastructure GNS3.

Ce volume peut occuper une activité intellectuelle de 16 à 35 heures, voir plus.

Dans une première partie, on tentera d'acquérir les fondamentaux des technologies des réseaux tels que les modèles TCP/IP et OSI, les composants et les concepts d'architecture des réseaux d'accès LAN/WAN, dans le *Data Center* et dans le nuage (*cloud*) et, enfin, les rudiments pour manipuler les identifiants codés en décimal, en binaire et en hexadécimal.

Dans une seconde partie, on aborde la manière de se connecter aux périphériques Cisco comme des commutateurs (*switches*) ou des routeurs (*routers*) et à comprendre l'environnement en ligne de commande Cisco IOS. Cette partie n'est pas directement vérifiée dans l'examen, mais elle est indispensable pour entrer dans les opérations de configuration et de diagnostic dans les environnements réels ou simulés.

La troisième partie s'intéresse à la couche Internet en général, aux adresses IPv4 et aux masques de sous-réseau, au NAT, aux protocoles ICMP, ARP, UDP et TCP. A titre de diagnostic, on proposera plusieurs commandes de prise d'information et de l'observation de trafic TCP/IP.

La dernière partie de ce volume "fondamental" serait incomplet sans parler d'IPv6. IPv6 est un sujet fortement vérifié dans les certifications Cisco CCNA. Cette partie s'intéresse à la reconnaissance et à la validation des adresses IPv6, à leur configuration sur les interfaces, à leur vérification et à leur diagnostic. Enfin, on trouvera un propos sur la manière de concevoir des plans d'adressage IPv6.

Première partie Fondamentaux des réseaux

Les données que nos ordinateurs transmettent à travers les réseaux sont transportées sous la forme de signal binaire grâce à des protocoles de communications qui enveloppent et développent les informations originales en différentes couches. Des **protocoles LAN/WAN**, des **protocoles de la pile TCP/IP** et leurs **modèles de communication** rendent ces échanges possibles d'un point d'extrémité du globe à un autre de manière efficace, sécurisée et peu coûteuse.

Certains **périphériques** tels que des commutateurs (*switches*) ou des routeurs (*routers*), des pare-feu (*firewalls*) ou encore des contrôleurs de réseaux locaux sans fil (*Wireless LAN controllers*) ainsi que les liaisons qui les interconnectent sont des composants matériels constitutifs des infrastructures de communications. Ces éléments construisent nos réseaux locaux domestiques et d'entreprise, de surveillance, de gestion, de paiement, nos centres de données, nos services à la clientèle, etc. et participent au déploiement de l'Internet. On représente les périphériques et leurs interconnexions par des **topologies** en forme de diagrammes. Les infrastructures de communication sont élaborées sur base d'**architectures** visant la robustesse, la sûreté et la haute disponibilité des données à travers le réseau grâce à des **modèles de conception**.

Les données sont transmises physiquement sous forme d'ondes électromagnétiques, sur des supports en cuivre (câble à paires torsadées, coaxial) ; sous forme d'ondes lumineuses, sur des supports comme l'air sur de courtes distances (ondes infra-rouges ou ultra-violettes) ou la fibre optique (laser/LED) ; sous forme d'ondes radio, à travers les airs sur de longues distances. Alors que nous avons l'habitude de parler en langues naturelles et à représenter les valeurs en base décimale, les ondes émises par les ordinateurs représentent des "bits de données" **codés en binaire ou en hexadécimal**.

Les services Internet fonctionnant avec la pile TCP/IP sont accessibles grâce à des technologies d'accès au réseau local (de type LAN) ou de type distant (pour des accès Internet, des inter-connexions de sites distants d'entreprise ou de centres de données ou encore avec des facilités VPN).

1. Protocoles et modèles de communication

Les protocoles du réseau peuvent être modélisés et catégorisés selon divers critères. On trouvera dans ce premier chapitre les principes fondamentaux sur les modèles de communication et leurs protocoles : définition, catégorisation, rôles, mise en couches, encapsulation et désencapsulation. On évoquera aussi les modèles OSI et TCP/IP, les technologies LAN, WAN et dérivés sans fil ainsi que les modèles de conception.

1. Définition d'un protocole de communication

Un protocole de communication est un ensemble de règles qui rendent les communications possibles, car les intervenants sont censés les respecter.

Les protocoles définissent une sorte de langage commun que les intervenants utilisent pour se trouver, se connecter l'un à l'autre et y transporter des informations.

Les protocoles peuvent définir toute une série de paramètres utiles à une communication :

- des paramètres physiques comme des modulations, des types de supports physiques, des connecteurs ...
- le comportement d'un certain type de matériel,
- des commandes,
- des machines à état,
- des types de messages,
- des en-têtes qui comportent des informations utiles au transport.

2. Standardisation, régulation et interopérabilité

2.1. Organismes de standardisation

Les protocoles sont discutés et élaborés par des organismes de standardisation.

Les protocoles TCP/IP sont formalisés par l'[Internet Engineering Task Force \(IETF\)](#) dans des documents publics qui prennent le nom de RFC ("Requests For Comments"). On désigne ces documents par des numéros de références. Tous les RFCs ne sont pas nécessairement des standards. Leur statut au sein du processus de standardisation peut être : *Informational*, *Experimental*, *Best Current Practice*, *Standards Track* ou *Historic*.¹ L'IETF dépend de l'ISOC.

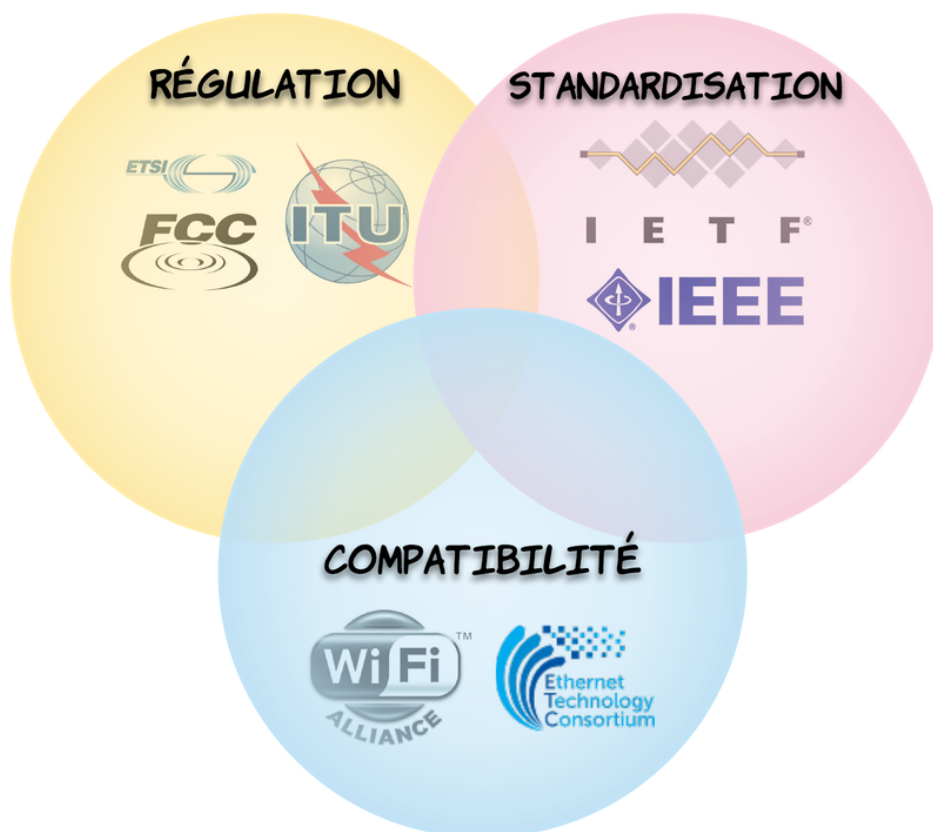
Les protocoles LAN / WAN / PAN² sont par contre formalisés par d'autres organismes comme l'[IEEE \(IEEE 802\)](#), l'[ITU](#), par l'[ANSI](#), etc.

2.2. Organismes de régulation et consortiums commerciaux

On prendra garde à distinguer ces organismes de standardisation de *consortiums commerciaux* comme la [WiFi Alliance](#) ou des *organismes étatiques nationaux et internationaux de régulation* comme le [FCC](#), l'[ETSI](#), l'[IBPT](#), ...

1. Pour un peu plus de détails sur les RFCs : fr.wikipedia.org et en.wikipedia.org.

2. Ces acronymes définissent des catégories de technologies physiques selon leur étendue : *Local Area Network* (LAN), *Wide Area Network* (WAN), *Personal Area Network*. Voyez plus bas.



Organismes de régulation, standardisation et compatibilité

Les organismes de régulation donnent les conditions d'usage des ondes sur l'espace public. Ce sont des organismes sous l'autorité des États qui s'entendent au niveau de l'Institution Internationale des Communications (ITU).

Les consortiums commerciaux tels que la "Wi-Fi Alliance" assurent que des matériels de fournisseurs respectant un standard comme IEEE 802.11 soient compatibles entre eux.

3. Nomenclature des protocoles

On peut catégoriser les protocoles sur base de différents critères comme la signalisation, la maintenance de la connexion, la fiabilité, selon les plans gestion, contrôle, donnée ou encore selon le type de technologie TCP/IP, LAN, WAN.

3.1. Signalisation

On désigne ici par "signalisation" tout message qui comprend des commandes utiles du protocole.

Un protocole à signalisation **In-Band** embarque la signalisation avec les données dans un même canal (HTTP).

Un protocole à signalisation **Out-of-Band** utilise une signalisation dans un protocole distinct (en téléphonie IP les protocoles SIP/SDP/RTP) ou un canal dédié (FTP).

3.2. Maintenance de la connexion

Un protocole **Orienté Connexion** est celui qui établit, maintient et ferme un canal au préalable de l'envoi des données (TCP, FTP).

Un protocole **Non Orienté Connexion** : Ethernet, IP, UDP, TFTP sont non orientés connexion et ne proposent aucun mécanisme de maintenance.

3.3. Fiabilité

Un protocole **Fiable** met en oeuvre des mécanismes de fiabilité tel que la reprise sur erreur, des accusés de réception, du contrôle de flux, etc. (TCP).

Un protocole **Non fiable** : Ethernet, IP, UDP, TFTP sont non fiables et ne proposent aucun mécanisme de fiabilité.

Les caractéristiques fiabilité et maintenance sont souvent associées, mais les deux caractéristiques peuvent être dissociées.

3.4. Plans

Les quatre plans “*Data*”, “*Control*”, “*Management*”, “*Services*” sont des concepts pour identifier le “plan” des opérations sur un routeur (L3) ou un commutateur (L2) ou dans une architecture.

Le plan **Donnée (*Data plane*)** fait référence aux **fonctions et processus de transfert de paquets**. On y trouve des protocoles transportant des données des utilisateurs (HTTPS, FTP, RTP), l’encapsulation de données. La qualité de service (QoS), le filtrage sont aussi sur ce plan.

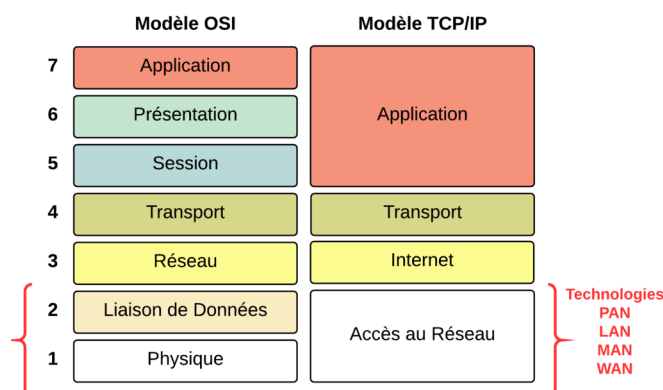
Le plan **Contrôle (*Control plane*)** concerne toutes les **fonctions de création des chemins** comme les protocoles de routage (OSPF, EIGRP, BGP) ou les protocoles LAN IEEE 802.1 (Spanning-Tree, Etherchannel, VLANs), les protocoles de redondance du premier saut FHRP (HSRP, VRRP, GLBP), etc. Ils émanent de l’infrastructure. Les protocoles de gestion d’infrastructure comme la résolution de noms (DNS), d’attributions d’adresses (DHCP, RA, DHCPv6), de voisinage (ARP/ND) peuvent être classés dans cette catégorie.

Le plan **Gestion (*Management plane*)** est le plan des **fonctions de gestion et de surveillance des périphériques**. On y trouve des protocoles d’accès distant (AAA, RADIUS, SSH), de supervision (NTP, SYSLOG, SNMP), de transfert de fichier (FTP, TFTP, SSH/SCP/SFTP). On pensera aussi aux interfaces “API” Rest supportées par HTTP et/ou HTTPS qui permettent de contrôler du matériel et des solutions de type NFV / SDN.

Le plan **Services (*Services plane*)** est un cas spécifique du plan Données (*Data plane*) pour du trafic qui demande une transformation comme par exemple l’encapsulation en tunnels GRE, MPLS VPNs, le chiffrement/déchiffrement TLS/IPsec, le mécanisme QoS, etc.

4. Modèles de communication

Un modèle de communication en couches organise les communications entre les hôtes sous forme d’enveloppement hiérarchique. Chaque couche remplit une fonction spécifique à la communication et dispose de caractéristiques propres. Chaque couche utilise un protocole, soit un ensemble de règles de communication.



Comparaison des modèles TCP/IP et OSI

La couche est enveloppée dans la couche sous-jacente. On parle alors d’encapsulation.

On utilisera principalement le modèle TCP/IP pour expliquer les communications au niveau logique. Le modèle OSI n’a plus d’intérêt que dans sa distinction entre la couche L1 et la couche L2 au niveau de la couche d’accès au réseau.

4.1. Avantages d'un modèle de communication en couches

Un modèle de communication en couches dispose de plusieurs avantages :

- Il permet de mieux comprendre le véritable fonctionnement des communications. Il est **utile à l'apprentissage**.
- Il est **utile au diagnostic** et au dépannage.
- Il permet **de développer ou d'adapter** de nouvelles fonctionnalités sans à revoir l'ensemble du modèle. Les modifications peuvent intervenir uniquement sur le protocole de la couche concernée.
- Il **favorise l'interopérabilité** entre différents matériels, technologies et constructeurs. Il favorise la croissance du marché dans le secteur des infrastructures et de services IT.

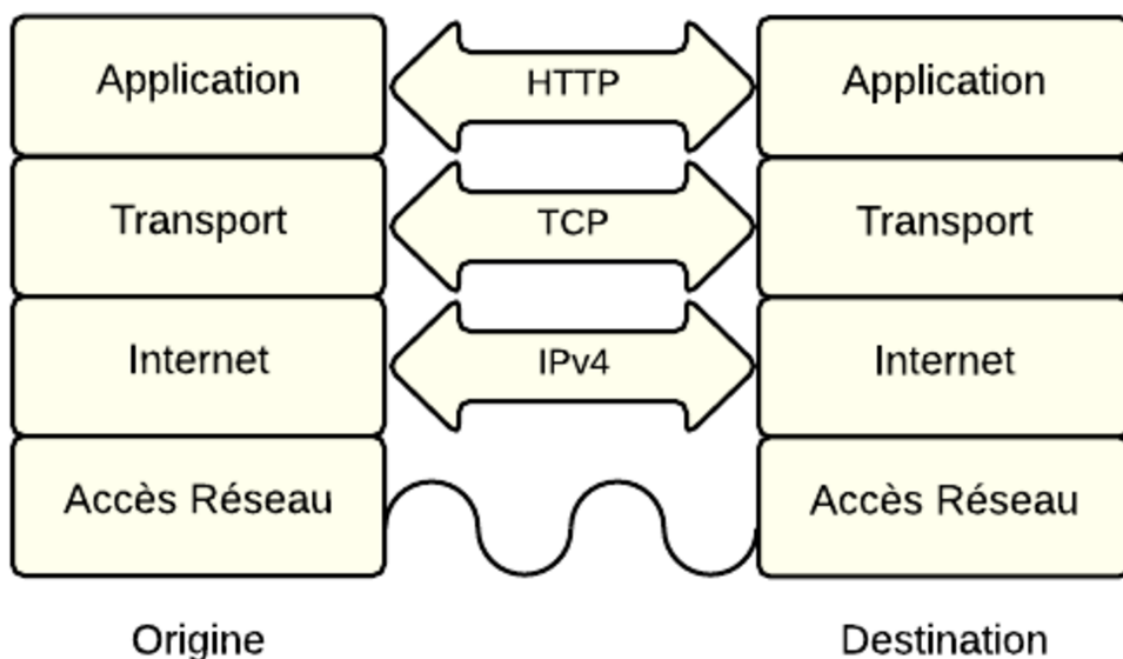
4.2. Modèles de communication utilisés dans ce document

On discutera dans ce document de quelques modèles de communication :

1. Le **modèle TCP/IP** qui implémente les technologies les plus courantes.
2. Le **modèle OSI** qui sert de référence académique et qui trouve des correspondances par rapport aux technologies TCP/IP, LAN et WAN.
3. Les **technologies LAN (Ethernet, Wi-Fi)** et les **technologies WAN (PPP, PPPoE, GRE, VPN)** qui disposent de leur propre modélisation au niveau des couches dites "basses", soit proches du signal physique dans les couches Physique (L1) et Liaison de données (L2).

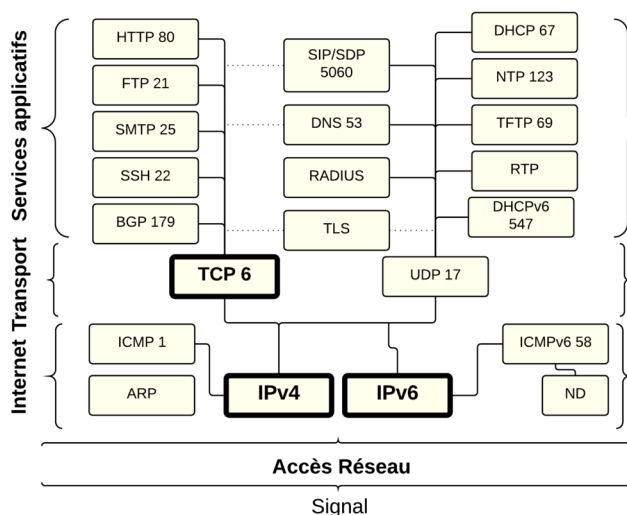
4.3 Protocoles TCP/IP

Les **protocoles TCP/IP** relèvent de la "logique", du *logiciel*, où d'une part, l'**Internet Protocol (IP)** identifie les noeuds de communication à travers l'interréseau et les meilleurs chemins, et où d'autre part, **TCP ou UDP** assurent la fonction de transport de bout en bout alors qu'un **protocole de couche application** rend un service de communication.



TCP/IP, communication d'égal à égal

C'est la technologie "Internet" qui utilise la pile des protocoles TCP/IP qui permet que des services comme Facebook, Google ou encore Amazon puissent fonctionner. C'est grâce à TCP/IP que nous pouvons tous les jours aller sur Internet.



Couches Internet, Transport et Application du modèle TCP/IP

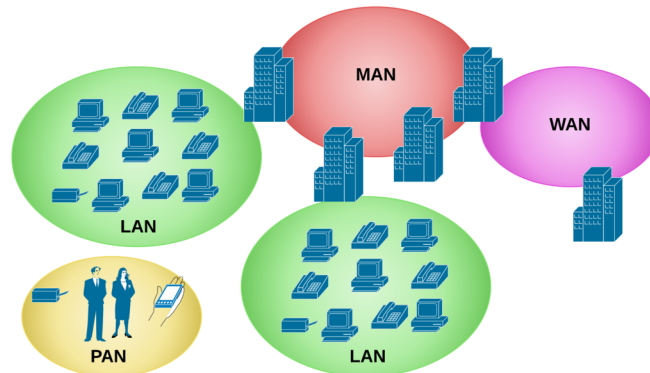
IP pour Internet Protocol, dans sa version 4 ou sa version 6, est celui qui permet d'identifier l'interface de communication de n'importe quels ordinateurs dans le monde et d'acheminer des paquets de données d'une extrémité du globe à une autre. TCP (Transmission Control Protocol) est celui qui permet de maintenir un canal fiable entre deux points IP. Son équivalent sans fiabilité et sans connexion est UDP (User Datagram Protocol). TCP et UDP transportent des protocoles applicatifs comme HTTPS ou DNS. Ces protocoles applicatifs sont utilisés par nos logiciels pour transporter des pages Web, des images, du son, bref n'importe quel type de données.

4.4. Protocoles LAN et WAN

Les *technologies et protocoles LAN/WAN* relèvent du *matériel/physique*. Elles assurent le **transport et la livraison physique** des données sur les liaisons locales avec les technologies :

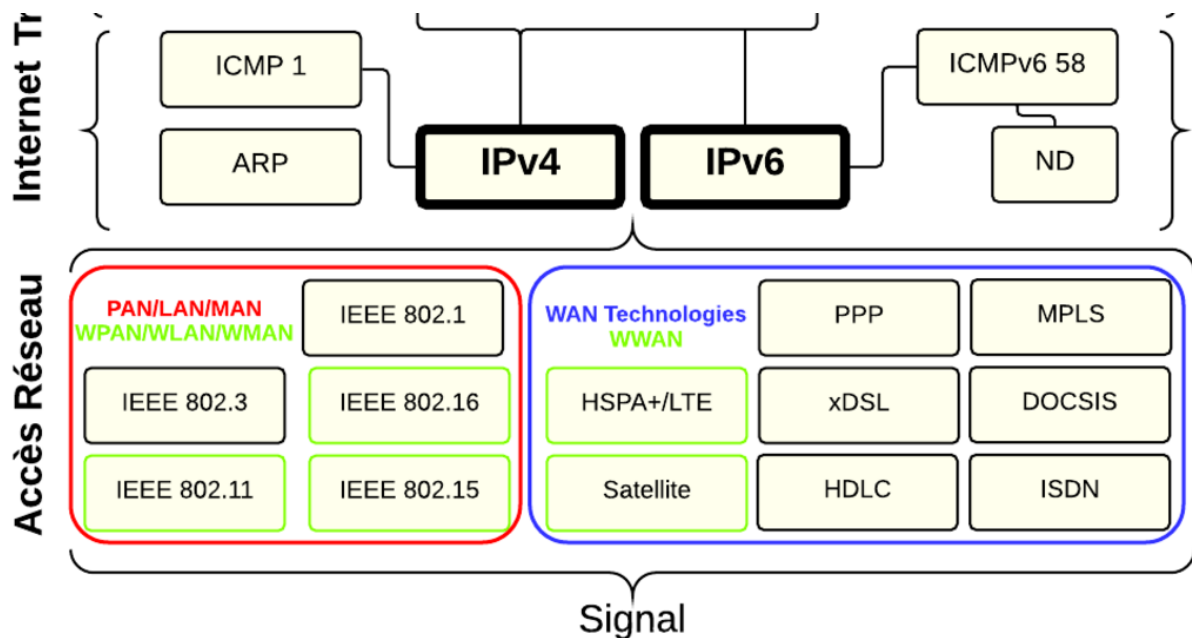
- **LAN** : Technologies au sein des réseaux locaux (confinés localement, rapides et permanents) : Ethernet, Wi-Fi, Bluetooth, WiMax, etc.
- **WAN** : Technologies qui interconnectent les routeurs, points intermédiaires de connectivité, plus lentes et couvrant de longues distances : xDSL/PPPoE, Metro-Ethernet, DOCSIS, LTE, 3G/4G/5G, Satellite, HDLC, Frame-Relay, ATM, ... mais aussi les connexions VPN.

Ces technologies disposent de leur qualification “Wireless” avec WPAN, WLAN, WWAN.



Technologies LAN, WAN, PAN, WLAN, WPAN, WWAN

Les technologies LAN, WAN, PAN, WLAN, WPAN, WWAN sont dites des **technologies d'accès**, car elles permettent d'accéder aux réseaux qui autorisent des connexions de bout en bout d'une extrémité à l'autre du globe, *a priori* en TCP/IP.



Couches physiques et Liaison de données

Si cette distinction fondamentale LAN/WAN tient essentiellement à la portée des technologies ou à leur usage, ce sont aussi d'autres critères comme l'origine de leur standardisation qui les opposent dans les rôles ou leurs objectifs.

5. Modèles de conception

Cisco Systems a contribué à la popularité des modèles de conception des réseaux. Ces modèles sont des guides de déploiement des infrastructures du réseau qui permettent aux organisations de rencontrer des objectifs de

performance et de disponibilité. On parle aussi d'architectures ou de topologie ou encore de *design* dans ce cadre. Les modèles de conception représentent la mise en oeuvre opérationnelle des protocoles du réseau dans l'infrastructure physique afin de contrôler au mieux le trafic et atteindre le niveau de service attendu par l'entreprise.

5.1. Modèles de conception traditionnels

On avait jusqu'en plus ou moins 2016 des modèles de déploiement respectant des architectures traditionnelles avec un modèle de conception hiérarchique et modulaire à trois couches : Access, Distribution et Core. Chaque couche assurant des rôles spécifiques sur l'objectif, les protocoles autorisés, les limites de la couche 2 et la couche 3, la sécurité, etc. Le chapitre ultérieur intitulé "[Principes de conception LAN](#)" évoque le sujet plus amplement.

Bien souvent, ces infrastructures sont difficiles à gérer et à faire évoluer. Elles demandent des interventions manuelles répétitives. Une vue unifiée de la surveillance et de la sécurité sont alors un véritable enjeu dans la gestion quotidienne de l'infrastructure ainsi que les reprises sur erreurs et la gestion des incidents.

Les architectures qui existent depuis longtemps deviennent difficiles à faire évoluer en demandant beaucoup d'interventions manuelles. Il n'est pas étonnant que l'environnement aie un impact sur les mentalités des professionnels en charge des infrastructures réseau.

Ces modèles sont toutefois toujours d'application dans les réseaux d'entreprise, ne fut-ce qu'en surcouche (*overlay*) d'une infrastructure physique sous-jacente – qui pourrait être conçue d'une autre manière.

5.2. Intent-Based Networks

"**Intent-Based Networks**" (IBN) est une nouvelle perspective de la gestion des réseaux dont l'acronyme est typiquement sorti des équipes marketing de chez Cisco. Les objectifs "Business" de l'organisation devraient être traduits en "politiques" du réseau par les professionnels de l'informatique de telle sorte que le réseau puisse atteindre ces objectifs **de manière automatisée et à grande échelle**, à titre d'"intention".

Il s'agirait en fait de se rapprocher de plus en plus d'architectures offrant du "**Network as a Service**" (NaaS). IBN demande une autre approche de la gestion de l'infrastructure avec d'autres interfaces que la ligne de commande (CLI) telle que la gestion à travers d'**APIs HTTP REST** avec des langages de programmation comme Python et des retours présentés en format **JSON**. Ces technologies sont déjà disponibles grâce à des déploiements de type **SDN** ("**Software Defined networks**") qui utilisent des **contrôleurs** pour centraliser la gestion de l'infrastructure.

Dans les offres des technologies WAN, on trouvera de plus en plus du "**Software Defined WAN**" (**SD-WAN**) voyant la connectivité WAN comme un service à la demande, prêt à la production et à l'évolutivité, comme par exemple commander à la demande des liaisons WAN entre deux villes d'un pays de manière "élastique". Son équivalent dans le LAN/WLAN (réseau local filaire et sans-fil) SD Access (SDA) apparaît dans les offres des fabricants. Avec SDN, le plan de contrôle et le plan data sont clairement distingués de telle sorte que les pratiques de gestion s'en trouvent modifiées.

Cette nouvelle vision du réseau a déjà eu pour conséquence de changer les infrastructures réseau du centre de données avec les concepts de fabrique (Fabric), de **topologie "Spine/Leaf"** (à deux niveaux) ou encore avec l'implémentation de protocoles comme VxLAN et l'intégration des réseaux SAN (de stockage, *Storage Area Network*, qui se distingue du LAN).

Aussi, avec IBN, on trouvera une multitude de propositions de déploiement du plan "contrôle" : dans le nuage, en local, embarqué sur les périphériques, etc. Les accès au réseau sans-fil devraient mieux s'intégrer aux infrastructures locales filaires, le "contrôle" pouvant disposer d'une vue abstraite des accès avec des politiques de sécurité unifiées.

Enfin, une nouvelle pratique fait son apparition avec celle de "**Network Assurance**" dans laquelle on retrouve des exigences de surveillance en temps réel avec des protocoles traditionnels, du *reporting* et des réponses automatiques. Concrètement, cela signifie que de nouvelles compétences en **programmation**, en **automation** et en gestion d'**infrastructures virtuelles** (locales ou dans le nuage) devront s'ajouter aux compétences des équipes en charge des réseaux, en plus de la connaissance des **nouvelles architectures** qu'ils devront gérer (ils n'en seront pas nécessairement les concepteurs initiaux).

6. Éléments clés à retenir

Les éléments clés à retenir concernant les protocoles et modèles réseau sont les suivants :

1. le rôle d'un protocole de communication réseau
2. les critères de distinction des protocoles (signalisation, connexion, fiabilité, plans)
3. la distinction entre les plans "donnée", "contrôle" et "gestion"
4. la distinction entre technologie d'accès et technologie Internet
5. la distinction entre technologies d'accès LAN et WAN
6. le concept de modèle en couche et d'encapsulation
7. le rôle des modèles de communication comme TCP/IP ou OSI
8. les modèles de conception de réseau
9. les concepts d'"Intent-Based Networks" (IBN) et de "Software Defined networks" (SDN)
10. le concept de topologie "Spine/Leaf"

En conclusion, un modèle de communication est une référence à maîtriser si l'on désire comprendre les technologies des réseaux.

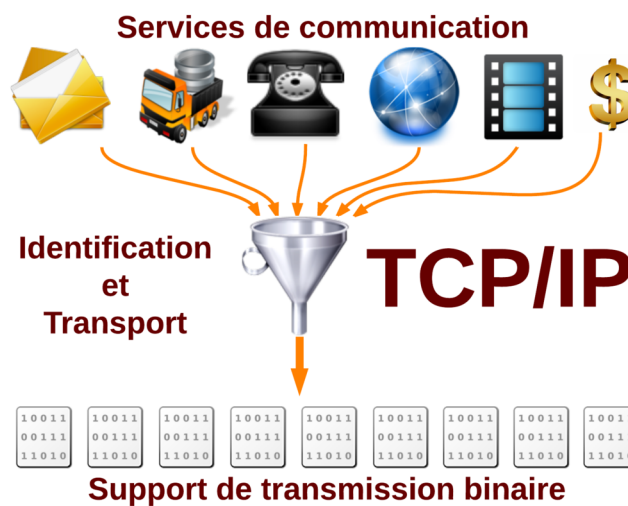
2. Modèles TCP/IP et OSI

Le chapitre “[Protocoles et modèles de communication](#)” expose les principes généraux des protocoles et modèles de communication. On trouvera ici une explication d’un second niveau sur les mécanismes de communication TCP/IP en regard du modèle OSI.

1. Modèle TCP/IP

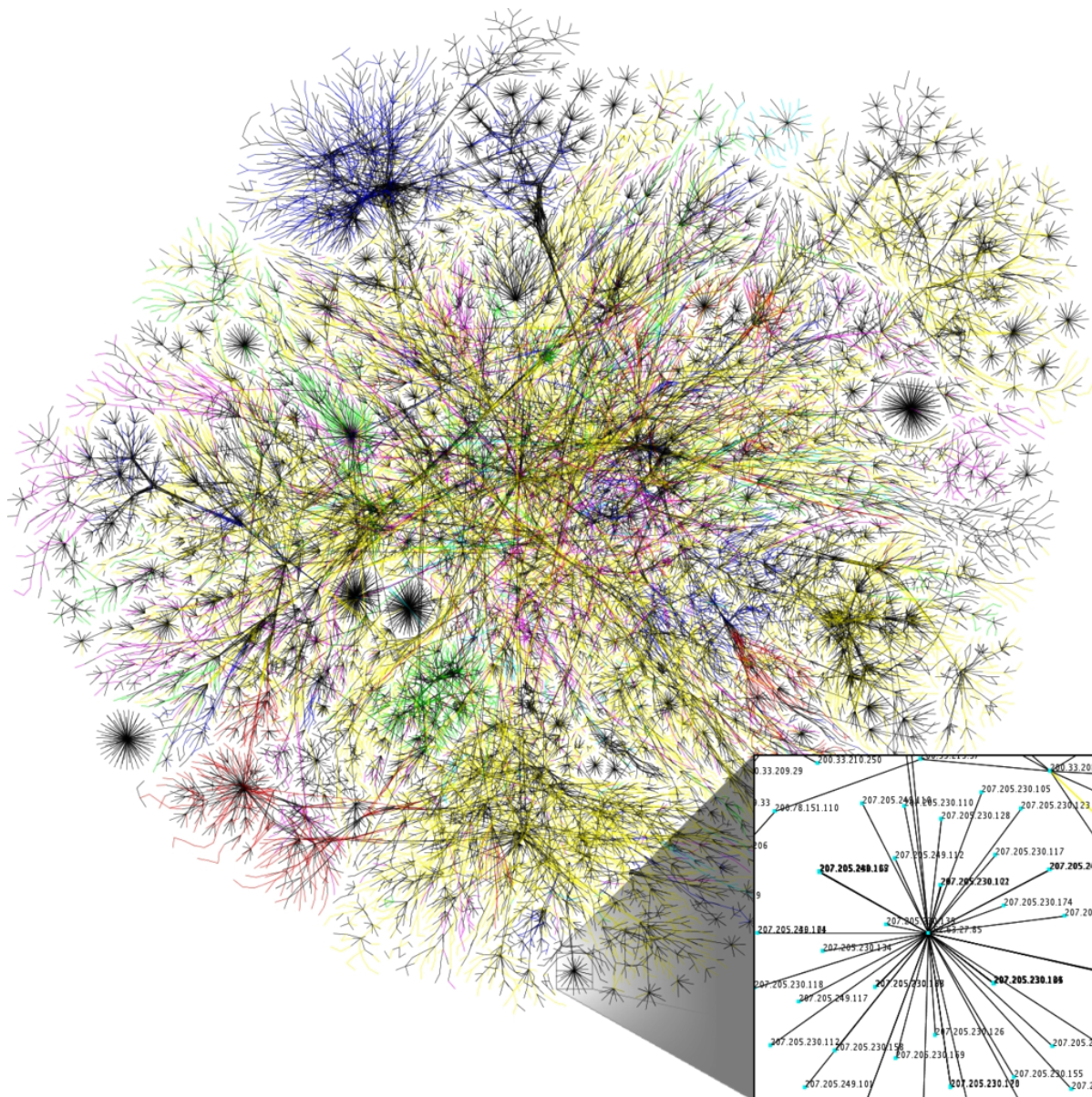
1.1. L’Internet

L’Internet peut encore représenter pour certains d’entre nous un nouveau paradigme : toute une série d’actes physiques courants est désormais accomplie à distance, manuellement ou automatiquement, par des humains avec des humains, mais aussi avec des machines et des machines entre elles. Recevoir ou envoyer du courrier, consulter des dossiers, établir des appels téléphoniques ou vidéos, aller en bibliothèque, regarder un film ou encore effectuer des transactions bancaires sont des oeuvres facilitées par les technologies de l’Internet.



Entonnoir TCP/IP

Mais de notre point de vue, l’Internet est l’**interconnexion de réseaux** à l’échelle du globe. En technologie IPv4, l’Internet a techniquement atteint sa taille limite de croissance !



Visualisation des multiples chemins à travers une portion d'Internet.[opte-project] Par The Opte Project (<https://commons.wikimedia.org/w/index.php?curid=1538544>). Carte partielle d'Internet, basée sur les données du 15 juin 2005 situées à opte.org. Chaque ligne lie 2 noeuds, représentant 2 Adresses IP. La longueur de chaque ligne indique le délai entre ses 2 noeuds.

Source de l'image

Qu'est-ce que l'Internet ?

Est-ce celui d'IPv4 ou d'IPv6 ? Est-ce la neutralité du réseau ? Qu'en est-il des répartiteurs de charge et des technologies en nuage (technologies *cloud*) ? Pour [Leslie Daigle de l'ISOC \(Internet Society\)](#), aucune de ces descriptions n'est très utile puisqu'elles sont toutes transitoires. Selon elle, une façon plus utile de penser l'Internet est de le caractériser par des **propriétés immuables**, c'est-à-dire qui ont résisté à l'épreuve du temps. Ces "**invariants**" comme les appelle l'Internet Society comprennent :

- sa portée mondiale,
- son objectif général,
- son accessibilité,

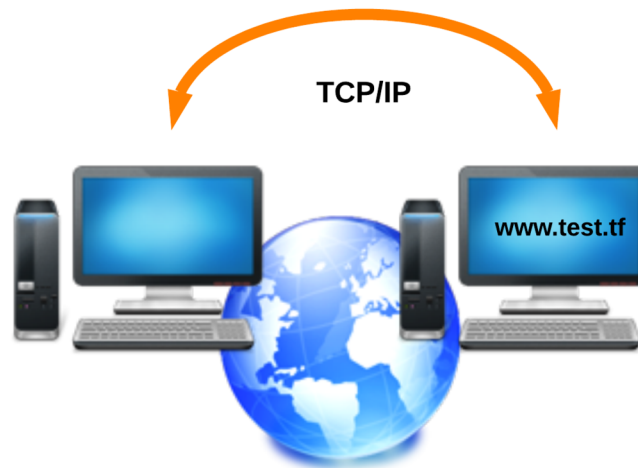
- son interopérabilité
- et sa technologie de blocs de construction réutilisables.
- Plus important encore, Internet est marqué par la collaboration.

Source : [Propos de Leslie Daigle de l'ISOC \(Internet Society\) recueillis par Marcia Savage, The Internet's Lessons in Resiliency, 16/05/2018.](#)

1.2. Objectifs de TCP/IP

Quels sont les objectifs de TCP/IP ?

- **Communiquer**
 - à l'échelle du globe
 - de manière libérale (ouverte)
- **quel que soit**
 - le contenu
 - le support
 - les hôtes
- **de manière robuste**

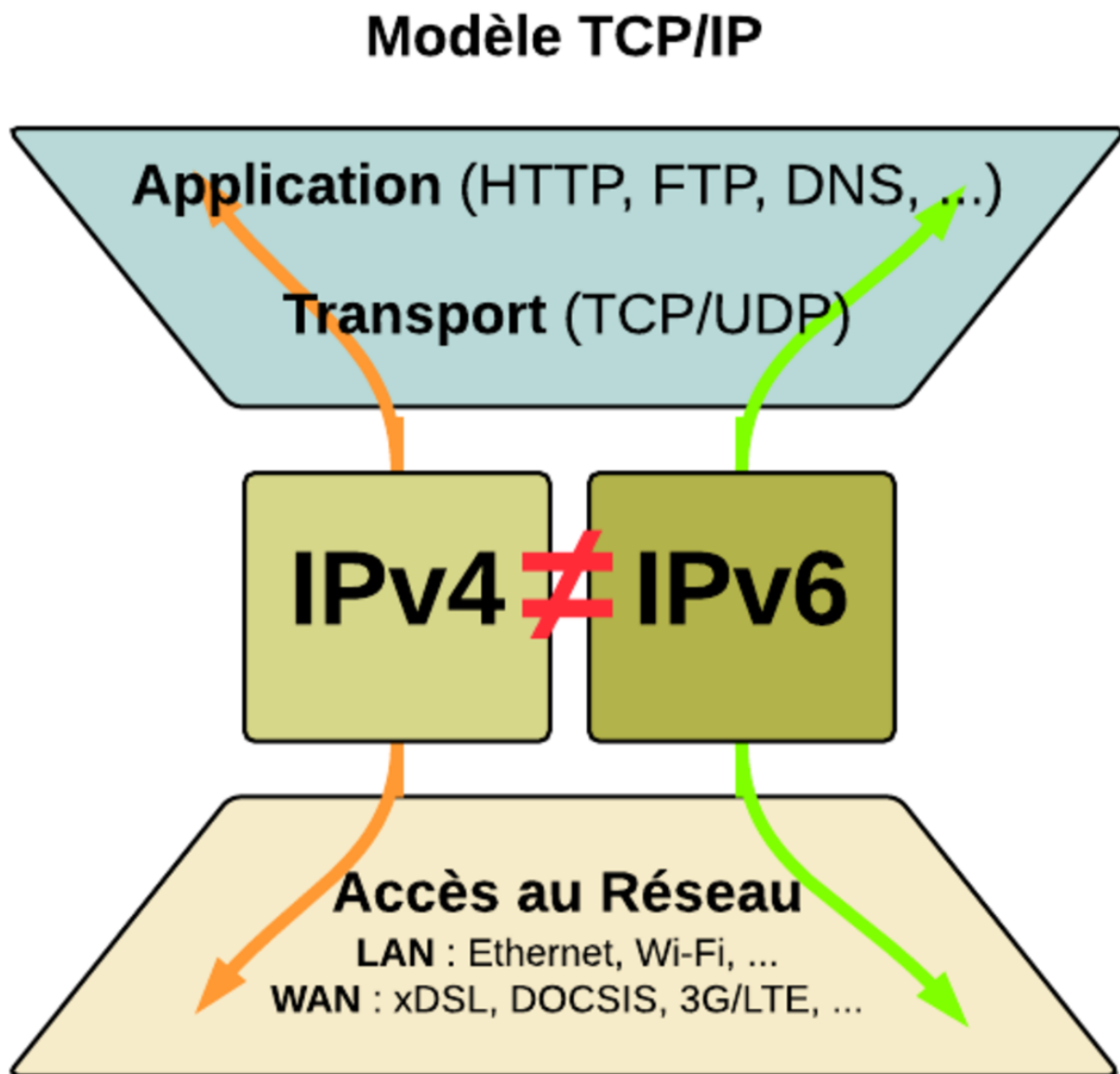


Communications TCP/IP

1.3. Modèle TCP/IP

Le modèle TCP/IP est fondé sur **quatre couches** qui enveloppent les messages originaux avant qu'ils soient placés sur le support physique sous forme d'ondes représentant les données de la communication.

Chaque couche assure une fonction de maintenance et de service de la communication. TCP/IP ne se préoccupe pas du contenu (les propos tenus par les utilisateurs dans les messages); il se contente d'assurer des fonctions qui facilitent les communications, le partage et la diffusion des informations.



Sablier TCP/IP

2. Modèle TCP/IP à quatre couches

2.1. Couche Application

- Elle est la couche de communication qui s'interface avec les utilisateurs.
- Exemples de protocoles applicatifs : HTTP, DNS, DHCP, FTP, ...
- Elle s'exécute sur les machines hôtes.

2.2. Couche Transport : TCP

- Elle est responsable du dialogue entre les hôtes terminaux d'une communication.
- Les applications utiliseront TCP pour un transport fiable et UDP sans ce service.

- Les routeurs NAT et les pare-feux opèrent un filtrage au niveau de la couche transport.

2.2. Couche Internet : IP

- Elle permet de déterminer les meilleurs chemins à travers les réseaux en fonction des adresses IPv4 ou IPv6 à portée globale.
- Les routeurs transfèrent le trafic IP qui ne leur est pas destiné.

2.3. Couche Accès au réseau : LAN/WAN

- TCP/IP ne s'occupe pas de la couche Accès Réseau
- Elle organise le flux binaire et identifie physiquement les hôtes
- Elle place le flux binaire sur les supports physiques
- Les commutateurs, cartes réseau, connecteurs, câbles, etc. font partie de cette couche

Au sens du modèle TCP/IP la couche Accès Réseau est vide, car la pile des protocoles Internet (TCP/IP) est censée “inter-opérer” avec les différentes *technologies qui offrent un accès au réseau*.

Plus on monte dans les couches, plus on quitte les aspects matériels, plus on se rapproche de problématiques logicielles.

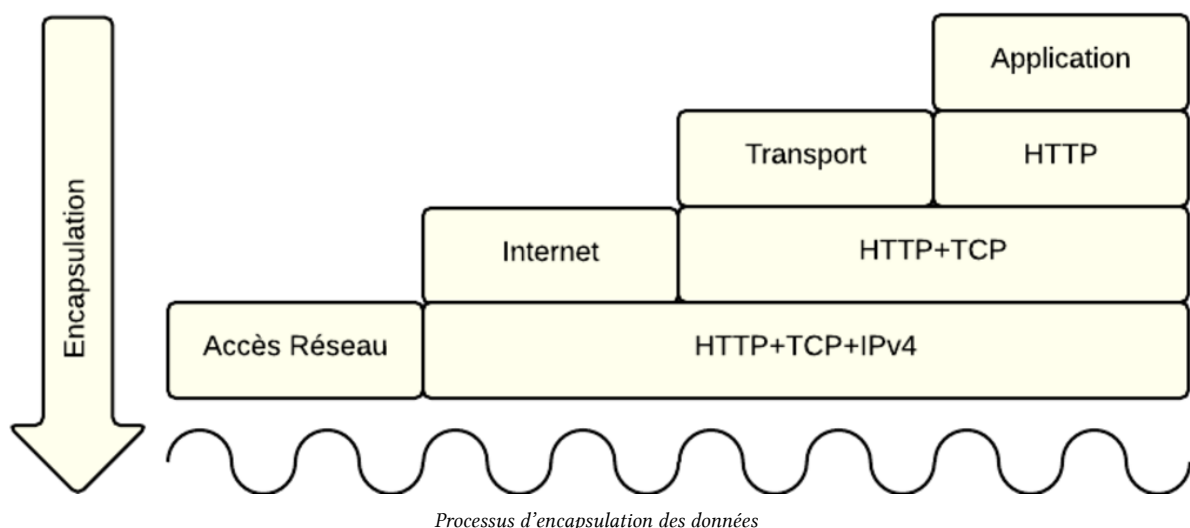
2.4. Encapsulation

Pour transmettre du contenu d'un ordinateur à un autre, l'utilisateur va utiliser un programme qui construit un message enveloppé par un en-tête applicatif, HTTP par exemple. Le message subit une première encapsulation.

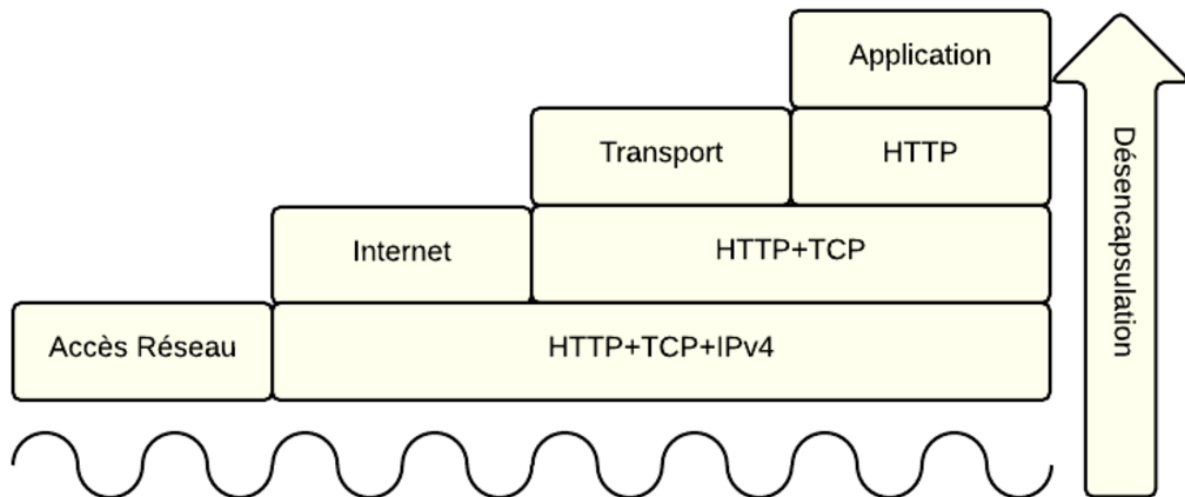
Le logiciel va utiliser un protocole de couche transport correspondant pour établir la communication avec l'hôte distant en ajoutant un en-tête TCP ou UDP.

Ensuite, l'ordinateur va ajouter un en-tête de couche Internet, IPv4 ou IPv6 qui servira à la livraison des informations auprès de l'hôte destinataire. L'en-tête contient les adresses d'origine et de destination des hôtes.

Enfin, ces informations seront encapsulées au niveau de la couche Accès qui s'occupera de livrer physiquement le message.



À la réception, l'hôte récepteur réalise l'opération inverse en vérifiant les en-têtes de chaque protocole correspondant à une des couches décrites. Ce processus s'appelle la désencapsulation.



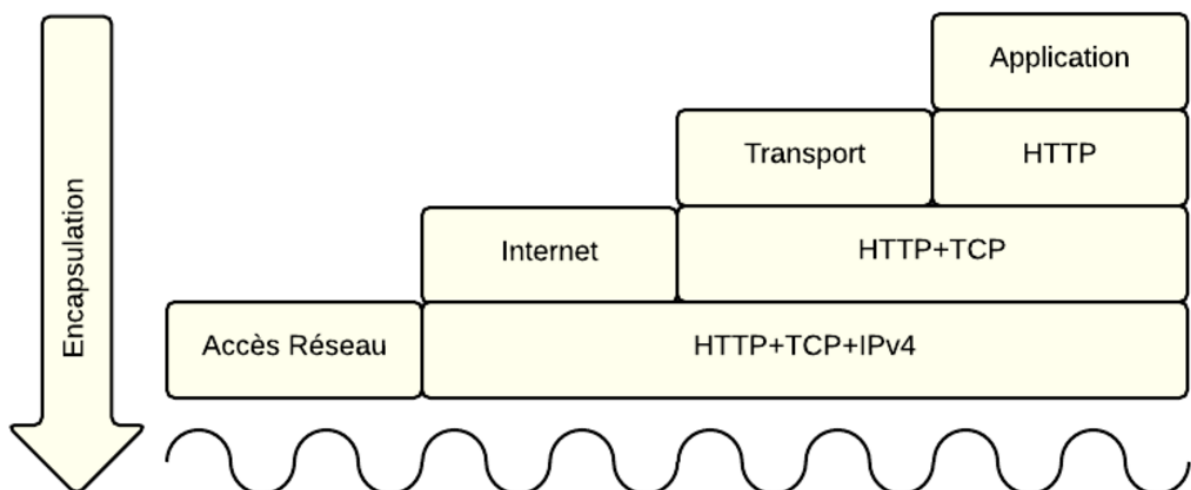
Processus de désencapsulation des données

Processus de communication

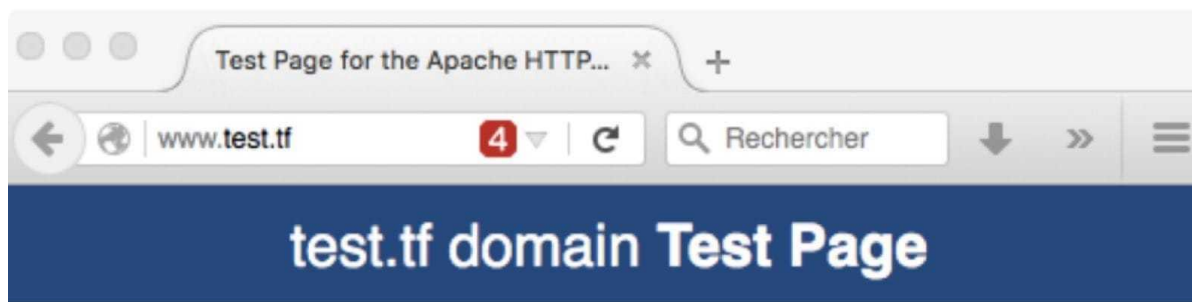
Chaque couche ajoute une information fonctionnelle au message original. À la réception, l'hôte examine chaque couche et prend une décision quant à ce trafic.

3. Illustration par un exemple

Rappelons le scénario de départ. Un utilisateur ouvre une page Web dans son navigateur et accède à sa page web favorite : <http://www.test.tf>. Sur le plan technique, une interface connectée au réseau émet un message original vers une destination distante (elle n'est manifestement pas locale) : il s'agit d'une requête HTTP qui demande à ouvrir une page Web (soit un fichier codé en HTML). En l'absence d'une page indiquée, c'est `index.html` qui sera servi.



Processus d'encapsulation des données



Requête HTTP GET sur un site Web

Moyennant une capture du trafic qui passe par l'interface, on peut observer les phénomènes qui seront expliqués plus bas.

3.1. Couche Application

Dans notre exemple, notre utilisateur ouvre sa page Web favorite et elle s'affiche immédiatement. Mais en réalité, avant d'obtenir ce résultat, il demande à lire une ressource (un fichier) située ailleurs, quelque part dans le réseau, sur `www.test.tf`. Dans le meilleur des cas, la plupart du temps, une page lui sera envoyée en retour et s'affichera dans son navigateur.

Il s'agit ici du premier message utile envoyé par l'utilisateur. Pour ce type d'usage ou d'application, il existe un protocole de **couche application** qui est largement disponible, à savoir par exemple HTTP¹. On pourrait résumer ce premier message par : "Serveur HTTP `www.test.tf`, donne-moi ta page `/index.html`".

On trouvera un grand nombre de protocoles de couche application, chacun offrant un type de service avec des caractéristiques propres. Ils sont développés dans le but d'offrir un service de communication qui soit au plus proche des utilisateurs (logiciels). Quelques exemples de commandes de couche application :

- Commandes **HTTP** : "donne-moi une page Web", "Voici la page", "donne-moi l'image", "Voici l'image".
- Commande **SIP** : "Tente d'établir une communication vocale ou vidéo à destination de tel numéro de téléphone, de tel courriel", "J'é mets une sonnerie", "j'ouvre la ligne"
- Commande **SMTP** : "livre ce message à `user@test.tf`"
- Commande **RTP** : "transmets telle capsule vidéo ou audio"
- Commandes **DNS** : "quelles sont les adresses IP de `www.test.tf`", "voici l'adresse IPv4 de `www.test.tf`"
- Commandes **DHCP** : "Donne-moi une adresse IP", "Voici des paramètres optionnels"

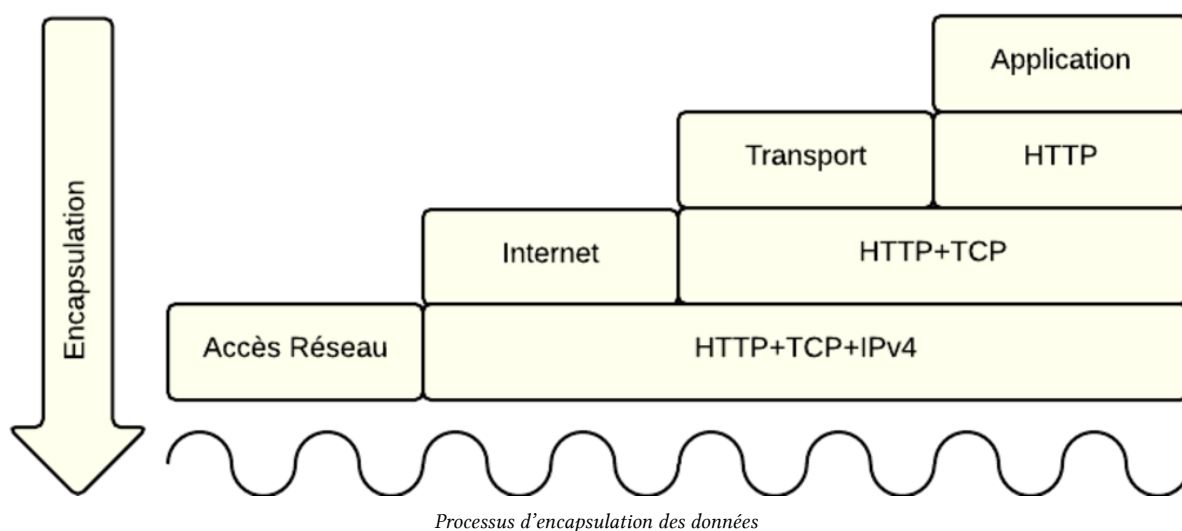
1. Aujourd'hui, on aura toujours une préférence pour les sessions sécurisées et vérifiées HTTPS supportées par SSL/TLS.

3.2. Couche Transport

Cette commande de couche application est alors enveloppée, “**encapsulée**” dit-on dans le jargon, par un protocole de **couche Transport** comme TCP. TCP apporte un service fiable pour que les utilisateurs puissent utiliser cette application Web. C’est justement TCP qui crée l’intuition d’une connexion directe entre les deux ordinateurs connectés, entre l’utilisateur et le serveur Web qui rend les pages à l’autre bout du monde.

Un canal de communication entre un “**port**” d’origine et un “**port**” de destination est établi entre les deux machines interlocutrices au préalable de l’envoi de la commande HTTP.

Si TCP fournit le canal de transmission entre les deux ordinateurs, il offre aussi une maintenance “connectée” de la communication. Il l’initie, la maintient avec des fonctionnalités telles que le contrôle de flux et la reprise sur erreur, et enfin il termine cette communication. On parle alors de protocole “**orienté connexion**”.



3.3. Couche Internet

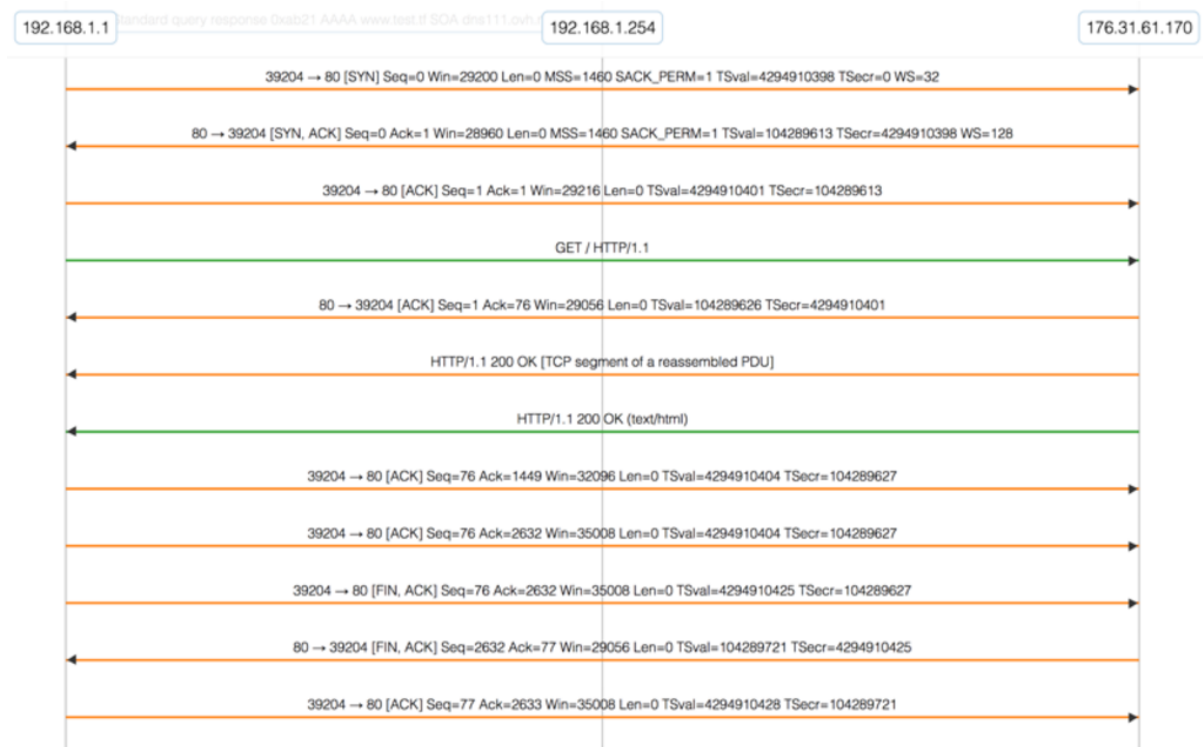
Pour que ces deux ordinateurs se reconnaissent de manière certaine, ils ont besoin d’autres identifiants que le seul nom de la destination ou les ports utilisés à l’origine et à la destination. À cet effet, une nouvelle encapsulation ajoute des informations de **couche Internet**, comme l’adresse IP d’origine (source) et l’adresse IP de destination. Elles identifient les hôtes de la communication d’un point de vue **logique** et **global**. C’est grâce à ces adresses que les données peuvent parvenir jusqu’au serveur de pages Web `www.test.tf` et que le trafic de retour est possible.

Des périphériques spécifiques qui interconnectent les lignes physiques se chargent de transmettre ces paquets d’informations jusqu’à la destination. Entre les deux ordinateurs, les paquets peuvent traverser plusieurs d’entre eux. Des routeurs relaient les paquets d’un point d’origine à un point d’extrémité de l’interréseau en fonction de l’**adresse IP de destination**.

3.4. Socket TCP/IP

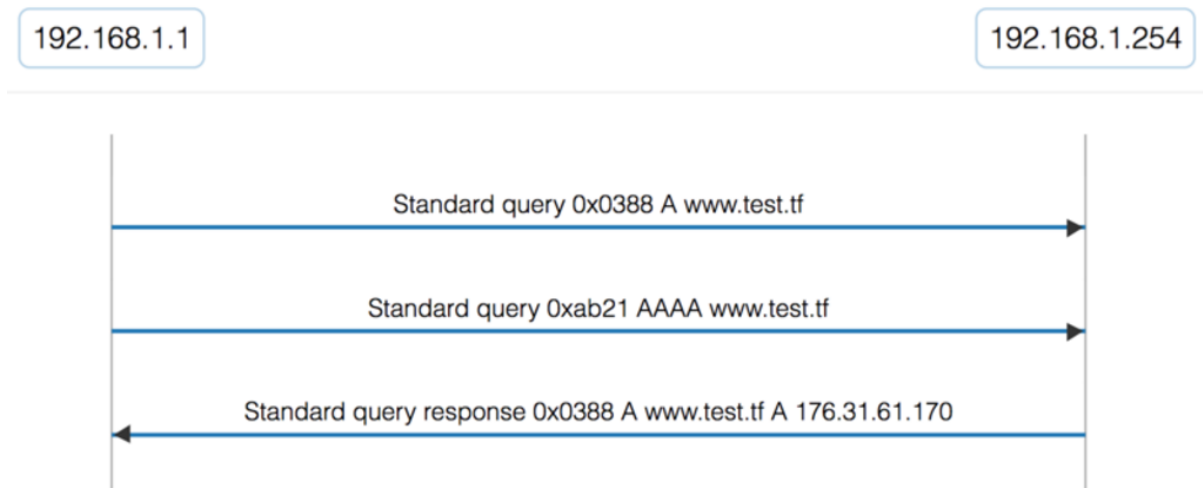
À partir d’un couple de type `adresse_IP :port_TCP` de la source et de la destination, soit des **sockets** dans le jargon, littéralement des “prises”, on crée une sorte de tunnel sur ces connexions ; on parlera alors de “**session**” TCP. TCP offre un **mode connecté** qui initie un dialogue, le maintient et puis le ferme entre les deux ordinateurs. Le canal logique qui permet d’acheminer les messages utiles (couche application) est ainsi créé. Donc en réalité, une connexion TCP aura été établie avant l’envoi de la requête HTTP.

Le *serveur* Web `www.test.tf` est un ordinateur dans le réseau dont le logiciel de service Web (Apache2 en l’occurrence) a mis le port TCP 80 de l’une de ses interfaces à l’écoute. Le port 80 est le port par défaut pour offrir un service Web en HTTP (TCP80). L’ordinateur qui émet la requête soit le *client* utilise un port aléatoire dans des valeurs élevées. Chaque protocole de couche application dispose d’un port par défaut (qu’il est utile de connaître), il est toujours possible de mettre à l’écoute un service sur un port non conventionnel.



[Flux HTTP TCP](<https://www.cloudshark.org/analysis/70eea568e03c/ladder>)

Si on est attentif à la capture, on constate que du trafic DNS (Application) supporté par UDP (couche transport) a demandé la résolution du nom en IPv4 (A Record) et en IPv6 (AAAA Record). UDP (User Datagram Protocol) est utilisé pour une communication de type “Raw”, à livraison brute, au contraire du service offert par TCP.



[Echange DNS en UDP](<https://www.cloudshark.org/analysis/70eea568e03c/ladder>)

3.5. Couche Accès Réseau

Faut-il encore que ces données puissent être physiquement transmises. Une nouvelle capsule place des informations de **couche Accès Réseau**. Par exemple, dans notre cas, des informations propres à la technologie *Wi-Fi* ou à la technologie *Ethernet* encapsulent le paquet IPv4. Ces dernières, **indépendantes des protocoles TCP/IP** permettent aux utilisateurs de placer physiquement toutes ces données sur un réseau local. Le réseau local prendra en charge le transfert de ces informations jusqu'à la bonne destination.

On peut trouver dans cette couche de nombreuses fonctions qui assurent le transport physique : identification,

livraison, contrôle du support, aspects physiques, connecteurs, entre beaucoup d'autres. En cela, cette couche se distingue des fonctions logiques et globales de TCP/IP. On retiendra aussi le rôle important que jouent les **commutateurs** Ethernet, "switches" dans le jargon, qui prennent leurs décisions de transfert sur base des **adresses MAC** codées dans cette couche.

Cette couche "basse" est aussi désignée comme étant le rassemblement de la couche Liaison de données (L2) et de la couche physique (L1) en référence au modèle OSI. On y place donc des protocoles autres que ceux de la pile TCP/IP : des protocoles IEEE 802 ou des normes ratifiées par l'ANSI, l'EIA/TIA ou encore l'ITU qui en fait disposent de leur propre modélisation jusqu'à la mise en ondes sur un support physique.

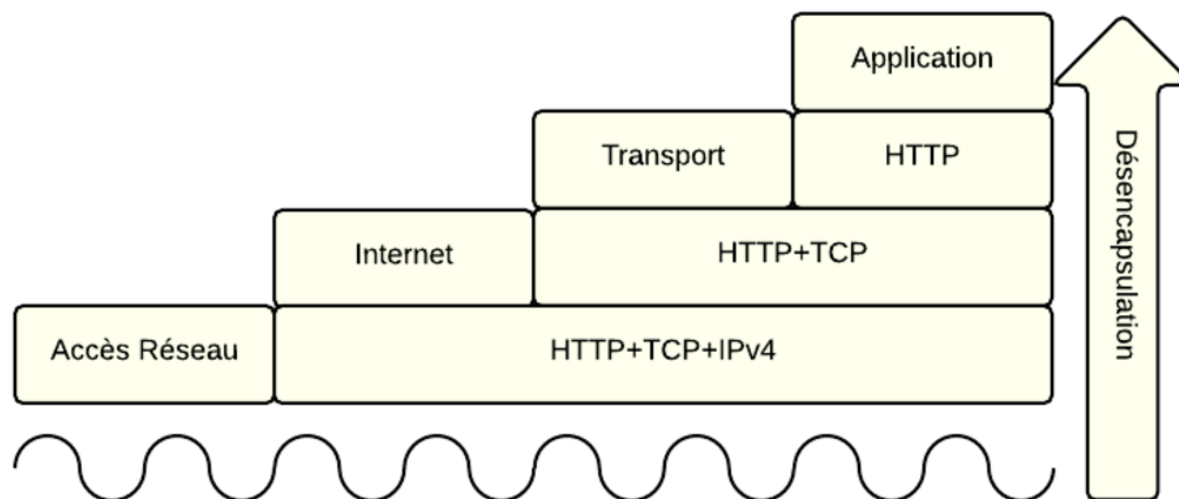
3.6. Résumé des opérations

On peut résumer cet exemple de transmission par le tableau suivant :

Couche	Protocole	Rôle de la couche	Matériel
Application	HTTP	Offre un service de communication utile : échanger des pages Web.	Ordinateurs
Transport	TCP	Offre un service fiable à HTTP, un port d'entrée/sortie et une session en mode connecté.	Ordinateurs
Internet	IP	Identifie les hôtes de manière unique et permet l'acheminement des paquets entre deux hôtes d'extrémité.	Ordinateurs, Routeurs
Accès Réseau	Wi-Fi	S'occupe de la livraison locale.	Ordinateurs, Commutateur, Point d'Accès, câblage cuivre/FO, paires torsadées, paire téléphonique, coaxial ondes EMI, RF, lumineuses

3.7. Désencapsulation du trafic à la réception

Du point de vue du serveur Web, soit à la réception du message, on imaginera que ce dernier examinera les informations de chaque couche, de la plus basse à la plus élevée. Il procédera à ce qu'on appelle une **désencapsulation**.



Processus de désencapsulation des données

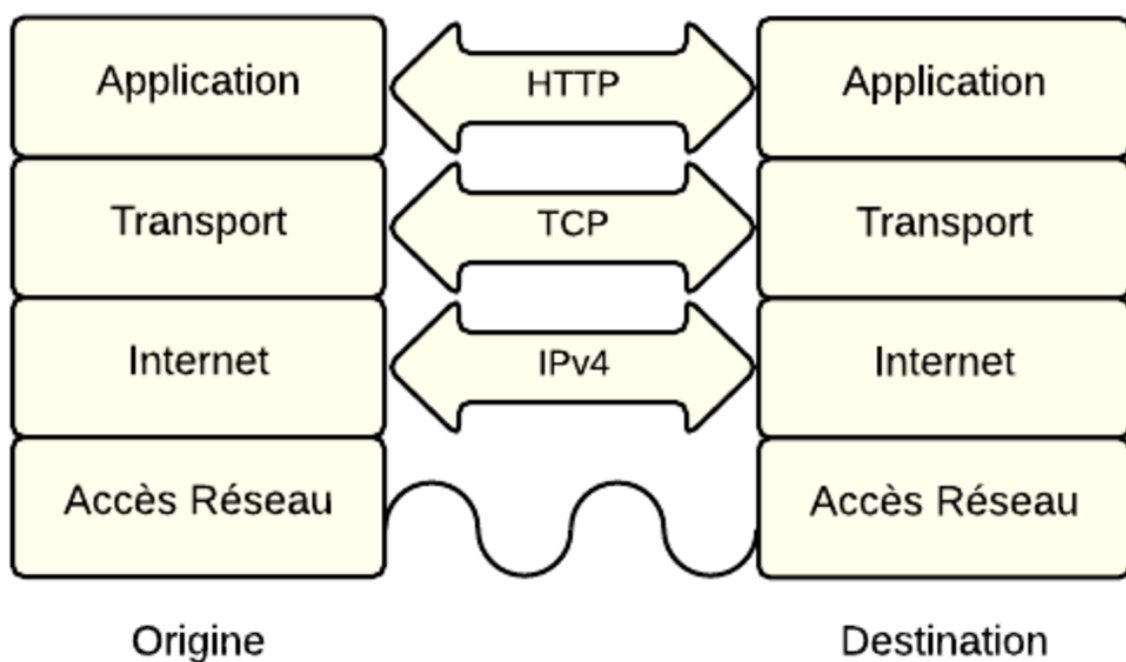
La machine de destination examine d'abord les informations de couche Accès Réseau et principalement l'adresse MAC de destination. S'il y trouve la sienne, il élimine les informations de cette couche et examine celle de la couche Internet et Transport. S'il y trouve son adresse IP et le port TCP en destination, il confie le message HTTP original à la couche Application qui est traité par un service système qui répond au protocole (Apache2 par exemple, /usr/sbin/httpd). Ce dernier peut à son tour offrir une réponse (une page Web) sur l'adresse et le port qui se trouve dans le message de la demande. Cette réponse est elle-même encapsulée par le serveur et désencapsulée à

l'arrivée chez le client.

3.8. Communication d'égal à égal

D'un point de vue holistique, c'est comme si les deux ordinateurs parlaient d'égal à égal, chaque couche assurant des fonctions spécifiques :

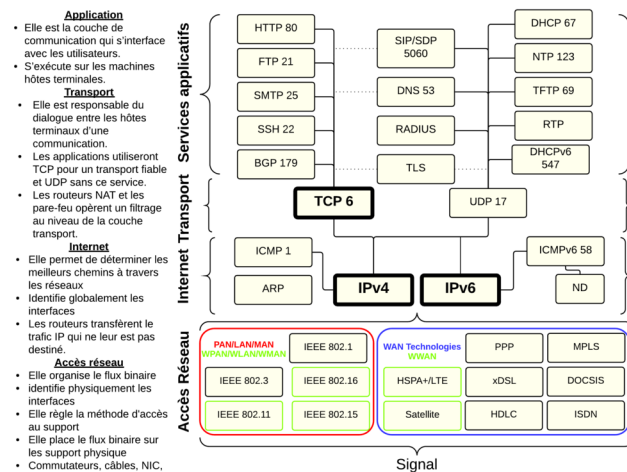
- *Application* : Message utile des utilisateurs
- *Transport* : Transport fiable ou non
- *Internet* : Identification et livraison globale
- *Accès Réseau* : Identification et livraison locale



TCP/IP, communication d'égal à égal

4. Modèles et protocoles en détail

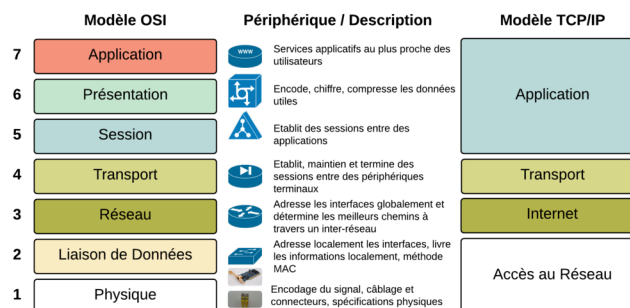
On sera attentif aux numéros de ports TCP et UDP associés aux protocoles de couche Application.



Modèle TCP/IP et protocoles en détail

4.1. Modèle OSI et comparaison au modèle TCP/IP

- La modélisation OSI propose un enveloppement en 7 couches
- Modèle théorique, académique
- Les couches sont aussi désignées par leur numéro
- On associera à chaque couche des rôles, des protocoles et du matériel.
- PDU : nom donné à l'encapsulation correspondante à une couche.



Modèle OSI et modèle TCP/IP comparés

4.2. Adressage, identifiants et matériel

Les machines et leurs interfaces disposent d'identifiants au niveau de chaque couche.

Couche	Identifiant	Exemple
Couche Application	Un protocole et un nom de domaine	<code>http</code> suivi de <code>www.cisco.com</code>
Couche Transport	Port TCP ou UDP	<code>TCP80</code> comme port par défaut pour HTTP
Couche Internet	Adresse IPv4 et/ou IPv6	<code>192.168.150.252/24</code> ou <code>2001:db8::1/64</code>
Couche Accès	adresse physique (MAC 802)	<code>70:56:81:bf :7c :37</code>

4.3. Tableau de synthèse

TCP/IP	OSI	Rôles	PDU	Protocoles	Matériel
Application	7 Application	Services au plus proche des utilisateurs	Données	HTTP, DNS, DHCP	Ordinateurs
Application	6 Présentation	encode, chiffre, compresse les données utiles	<i>idem</i>	<i>idem</i>	<i>idem</i>
Application	5 Session	établit des sessions entre des applications	<i>idem</i>	<i>idem</i>	<i>idem</i>
Transport	4 Transport	établit, maintient, termine des sessions entre des hôtes d'extrémité.	Segment	TCP ou UDP	Ordinateurs, routeurs NAT, pare-feux
Internet	3 Réseau	Identifie les hôtes et assure leur connectivité	Datagramme ou paquet	IPv4 ou IPv6	Routeurs
Accès Réseau	2 Liaison de Données	Détermine la méthode d'accès au support, organise les bits de données	Trame	Ethernet IEEE 802.3, Wi-Fi IEEE 802.11, pontage 802.1	Commutateurs, cartes d'interface réseau
Accès Réseau	1 Physique	s'occupe du placement physique du signal	bits	Normes physiques : xDSL (WAN), 1000-BASE-TX	Câblage (UTP CAT 6) et connecteurs (RJ-45), bande fréquences (2.4 GHz, 5 GHz)

5. Éléments clés à retenir

Les éléments clés à retenir sur le modèle TCP/IP sont les suivants :

- le principe et la portée du modèle TCP/IP
- le numéro de chaque couche (OSI), et ses protocoles, procédures, principes, matériels et PDU associés
- les principes d'adressage, identifiants et matériel au niveau des couches d'Accès Réseau, Internet et Transport
- être capable d'interpréter sommairement un capture de trafic sur base des quatre couche du modèle TCP/IP

3. Composants de base du réseau

Ce chapitre fondamental permet de mieux identifier les éléments qui composent un réseau (routeurs, commutateurs, concentrateurs, pare-feux, etc.) afin de les placer dans topologies d'exercices représentatifs ou de les placer logiquement en faisant à des modèles de conception.

1. Rôles des périphériques

Chaque couche, voire chaque protocole, dispose de sa propre vision des rôles des périphériques lors des procédures qui l'occupe.

IP voit deux rôles :

1. Les **hôtes terminaux** : nos ordinateurs au bout du réseau
2. Les **routeurs chargés** de transférer les **paquets** en fonction de l'**adresse L3 IP (logique, hiérarchique)** de destination. Ils permettent d'interconnecter les hôtes d'extrémité.

Aussi au sein du réseau local (LAN), le **commutateur** (switch) est chargé de transférer rapidement les **trames** Ethernet selon leur **adresse L2 MAC (physique)** de destination. Il ne s'occupe jamais des informations dites "L3", de couche 3 du modèle OSI comme les paquets ou datagrammes et les adresses IP qu'ils contiennent.

2. Les périphériques du réseau sont des ordinateurs

Les périphériques du réseau sont composés d'une partie matérielle :

- dédiée (sur des serveurs),
- embarquée (sur des plateformes autres que des serveurs de type Intel)

Cette partie matérielle est de plus en plus virtualisée notamment sur des **infrastructures de virtualisation** de type Intel X86. Un exemple est le [routeur virtuel Cisco Systems CSR-1000v](#) à déployer dans le centre de données (*data center*). Ces capacités de virtualisation sont disponibles sur des plateformes bien connues de type Intel avec leurs spécificités (CPU, périphériques, etc.).

Ces périphériques et leurs capacités matérielles sont exploités à partir d'un **système d'exploitation** :

- propriétaire comme Cisco IOS
- Open Source comme Linux ou BSD et leurs dérivés
- Basé Open Source avec une sur-couche propriétaire comme Cisco IOS-XE (Linux)

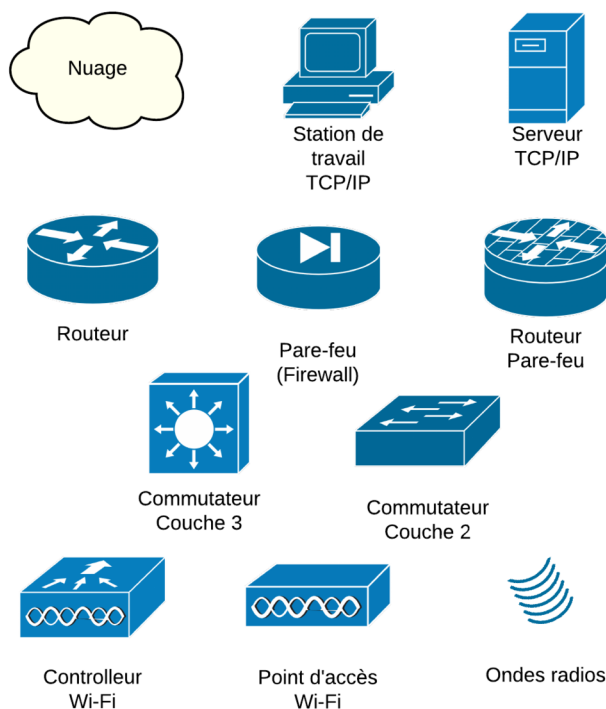
On prendra garde à distinguer les fonctions que les périphériques du réseau remplissent au niveau des **plans** :

- plan "donnée" : le périphérique transfère-t-il du trafic utilisateur ?
- plan "contrôle" : le périphérique se gère-t-il localement ou à partir d'un contrôleur ?

Enfin les périphériques peuvent être placés dans un niveau de conception dans une des couches "Access", "Distribution" ou "Core".

3. Composants d'un réseau associés aux couches OSI

On trouvera ici la représentation en forme de diagramme des composants de base d'un réseau.



Composants de base du réseau

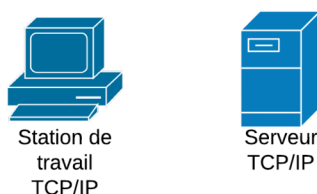
Composants du réseau

Chacun de ces périphériques est associé à une couche du modèle OSI avec "L" pour "*Layer*" :

Périphérique	Couche
Routeur (Router)	L3
Commutateur (Switch) L3	L2/L3
Commutateur (Switch) L2	L2
Pont (Bridge)	L2
Concentrateur (Hub)	L1
Répéteur (Repeater)	L1
Contrôleur WLAN	L2/L3/L7
Point d'accès sans-fil (AP) Wi-Fi	L1/L2
Carte réseau (NIC)	L2
Hôte terminal	L3/L4/L7

4. Périphérique terminal

Le périphérique terminal est l'hôte qui est situé à l'extrémité d'une communication : par exemple un poste client et un serveur sont des hôtes terminaux.

*Périphérique terminal*

Les périphériques terminaux accèdent au réseau via une technologie d'accès (L1/L2). Les périphériques clients peuvent être des ordinateurs de bureau, des mobiles ou des tablettes, mais de plus en plus des "objets" (IoT). Les serveurs sont ceux qui offrent et rendent des services à des clients. En général, ils sont situés dans des centres de données dédiés tels quels ou dans le "nuage".

5. Routeur (*router*)

On trouvera une présentation complète du rôle des routeurs, de leur fonctionnement et de leur format dans le chapitre "[Introduction aux routeurs Cisco \(lab\)](#)". Mais pour l'instant, tentons de comprendre son rôle fondamental.

Le **routeur** est ce matériel de couche 3 du modèle OSI (L3) qui :

- interconnecte des domaines réseaux IP différents ;
- transfère le trafic qui ne lui est pas spécifiquement destiné ;
- transfère le trafic IP grâce à sa table de routage vers les bonnes destinations, s'il les connaît.



Routeur

Symbole représentant un routeur

Sur le plan physique, les routeurs sont caractérisés par un nombre limité d'interfaces, mais d'une grande diversité (chez Cisco Systems), permettant à ce matériel professionnel de se connecter à n'importe quelle technologie d'accès. De plus chez Cisco Systems, ce matériel peut embarquer un bon nombre d'autres services du réseau comme par exemple des services de communication unifiée, de filtrage et de sécurité comme le pare-feu ou un système de détection/prévention d'intrusion (IDS/IPS) et même des capacités de virtualisation.

6. Commutateur d'entreprise (*switch*)

La partie "[Commutation Ethernet](#)" détaille amplement le rôle et le fonctionnement des commutateurs Ethernet. Essayons d'identifier le rôle du commutateur, le "switch", élément de couche 2 (L2) essentiel.



Commutateur
Couche 2

Symbole représentant un commutateur d'entreprise

Le commutateur d'entreprise (*switch*), typiquement de couche 2 liaison de données (L2) est ce matériel qui :

- interconnecte les périphériques terminaux du LAN pour un transfert local rapide ;
- transfère le trafic en fonction de l'adresse MAC de destination trouvée dans les trames et de sa table de commutation (*CAM table, ternary content addressable memory*), la prise de décision de transfert est réalisée au niveau matériel *hardware* grâce à des puces *ASIC* ;
- transfère le trafic *unicast* (à destination d'un seul hôte) uniquement sur le bon port de sortie ;
- transfère le trafic *Broadcast* (à destination de tous) et *multicast* (à destination de certains) est transféré par tous les ports sauf le port d'origine.

Dans les modèles de conception du réseau, le commutateur couche 2 (*switch L2*) fondamental remplit uniquement des fonctions L2 (transfert local, segmentation VLANs, marquage QoS, sécurité EAP/802.1x) et se place dans la couche "Access".

7. Commutateur multicouche (*Multilayers switch*)

Un **commutateur multicouche** (*Multilayers switch*) est un commutateur d'entreprise capable de remplir des tâches de routage et des services avancés. On l'appelle aussi "switch L3". Il s'agit d'un véritable routeur dédié au réseau LAN d'entreprise.



Commutateur
Couche 3

Symbole représentant un commutateur multicouche (L2/L3)

Ce type de matériel dispose de capacités améliorées en CPU/RAM, mais aussi et surtout en capacité de transfert matériel, en nombre d'interfaces et en technologies supportées. On les place souvent au niveau des **couches de conception** "Distribution" ou "Core". Le chapitre intitulé [Principes de conception LAN](#) offre plus de détails sur ce type de matériel.

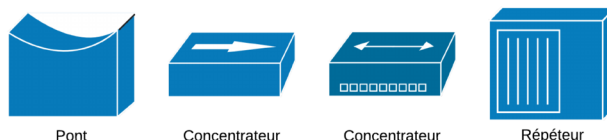
8. Ponts, Concentrateurs et Répéteurs

Les "Ponts", "Concentrateurs" et "Répéteurs" sont des éléments aujourd'hui plus conceptuels que réels (sauf dans certains déploiement en Wi-Fi).

Un **pont** (*bridge*) filtre le trafic entre deux segments physiques en fonction des adresses MAC. Le point d'accès Wi-Fi est une sorte de pont.

Un **concentrateur** (*hub*) est un périphérique qui concentre les connexions et étend le segment physique. À la différence du commutateur, le trafic sort par tous ses ports ; il ne prend aucune décision quant au trafic. En Ethernet, il était souvent identifié comme un répéteur multi-ports, ils ont été progressivement remplacés par des commutateurs de telle que ce matériel de "concentration" n'est plus fabriqué depuis longtemps.

Un **répéteur** (*repeater*) étend le signal entre deux ou plusieurs segments. Il ne prend aucune décision quant au trafic à transférer. On peut encore trouver des répéteurs dans les architectures Wi-Fi.



Ponts, Concentrateurs et Répéteurs

9. Matériel sans-fil

Avec des points d'accès légers (*Lightweight AP*), le **Wireless LAN Controller (WLC)** prend en charge les fonctions d'association ou d'authentification des APs, ces derniers devenant des interfaces physiques fournissant la connectivité. Le contrôleur fournit l'intelligence, la gestion, la configuration des APs. Il participe à une vue unifiée du réseau filaire et sans-fil.



Contrôleur
Wi-Fi



Point d'accès
Wi-Fi



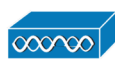
Ondes radios

Symbole représentant un point d'accès et un contrôleur WLAN

Un **point d'accès sans-fil** fournit le service du réseau sans-fil au sein d'une zone de couverture radio (cellule, *cell*). Un point d'accès dit "léger" remplira des tâches matérielles de chiffrement et de transport par exemple alors que l'intelligence sera assurée par un contrôleur.



Contrôleur
Wi-Fi



Point d'accès
Wi-Fi



Pont sans-fil



Ondes radios

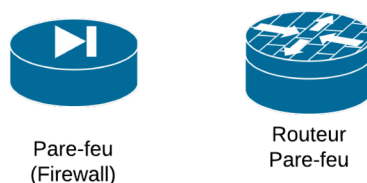
Symbole représentant un point d'accès, un pont sans-fil et un contrôleur WLAN.

Un réseau sans-fil contrôlé dispose de plusieurs avantages :

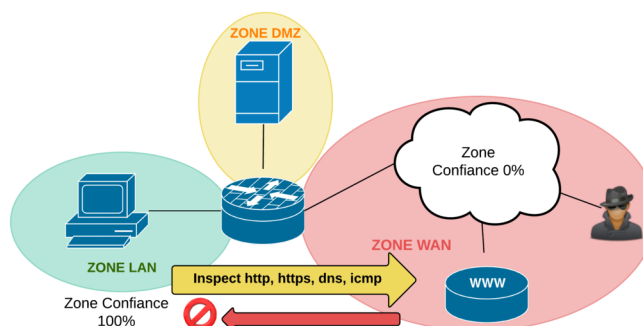
- **Fréquences Radio dynamiques** : Les points d'accès adaptent automatiquement la force du signal.
- **Processus de déploiement facilité** : Le contrôleur de réseau sans-fil (WLAN) assure la gestion centralisée des utilisateurs et des VLANs.
- **Performance des utilisateurs optimisée** : Un contrôleur de réseau sans-fil répartit la charge pour maximiser le transfert.
- **Processus de mise à jour facilitée** : Le contrôleur de réseau sans-fil (WLAN) peut déployer aisément des mises-à-jour sur les points d'accès.

10. Pare-feu

Un **pare-feu** (*firewall*) protège des tentatives de connexion directe venant d'un réseau comme Internet. Par contre, il laisse entrer le retour légitime du trafic initié d'une zone de confiance comme un LAN. Il tient compte de l'état des sessions de couche 4 établies (TCP, UDP, ICMP, etc.). On parle alors de pare-feu à état.



Symboles pare-feu firewall



Topologie pare-feu avec DMZ

Quelques éléments essentiels sont à retenir concernant les pare-feux en général :

- Dans un système d'information, les politiques de filtrage et de contrôle du trafic sont placées sur un matériel ou un logiciel intermédiaire communément appelé pare-feu (*firewall*).
- Cet élément du réseau a pour fonction **d'examiner et filtrer le trafic qui le traverse**.
- On aussi peut le considérer comme une **fonctionnalité** d'un réseau sécurisé : la fonctionnalité pare-feu.
- L'idée qui prévaut à ce type de fonctionnalité est le **contrôle des flux du réseau TCP/IP**.
- Le pare-feu limite le taux de paquets et de connexions actives. Il reconnaît les flux applicatifs.
- Il se place au sein du routage TCP/IP en connectant des réseaux distincts, il fait alors office de routeur.
- Il agit au minimum au niveau de la couche 4 (L4), mais il peut inspecter du trafic L7 (Web Application Firewall)
- Il ne faut pas le confondre avec le routeur NAT

On trouvera plus de détails sur les pare-feux dans les documents [Concepts Pare-Feux Firewall](#) et [Lab Cisco IOS Zone Based Firewall](#).

11. IPS

“IDS” et “IPS” sont des éléments connexes qui sont le résultat d'une évolution technologique. Un “IDS” détecte des intrusions et il devient “IPS” quand il est capable d'y réagir automatiquement.



Symbole d'IDS/IPS

Un système de détection d'intrusion (IDS) est un dispositif ou une application logicielle qui surveille un réseau ou des systèmes pour déceler toute activité malveillante ou toute violation de politique de sécurité. Toute activité malveillante ou violation est généralement signalée à un administrateur ou est recueillie de façon centralisée au moyen d'un système de gestion des informations et des événements de sécurité (SIEM). Un système SIEM combine des sorties provenant de sources multiples et utilise des techniques de filtrage des alarmes pour distinguer les activités malveillantes des fausses alarmes.

Les systèmes de prévention des intrusions (IPS), également connu sous le nom de systèmes de détection et de prévention des intrusions (IDPS), sont des dispositifs de sécurité réseau qui surveillent les activités du réseau ou du système pour détecter toute activité malveillante et **tentent de les bloquer ou de les arrêter**.

Les systèmes de prévention des intrusions peuvent être classés en quatre types différents : Système de prévention des intrusions en réseau (NIPS), Système de prévention des intrusions sans fil (WIPS), Analyse du comportement du réseau (NBA), Système de prévention des intrusions basé sur l'hôte (HIPS)

La majorité des systèmes de prévention des intrusions utilisent l'une des trois méthodes de détection suivantes : **l'analyse basée sur la signature, l'analyse statistique des anomalies et l'analyse du protocole dynamique**.

Snort, Sourcefire (Cisco Systems) et Suricata sont des exemples d'IDS/IPS bien connus.

On trouvera plus de détails sur les IDS/IPS dans le chapitre "[Concepts IDS IPS](#)"

12. Supports de transmission

Les supports de transmission des données interconnectent physiquement les périphériques et transportent les données sous forme d'ondes. Les données sont codées en ondes électromagnétiques (sur des supports métalliques comme des câbles en cuivre), en ondes lumineuses (à travers les airs ou à travers de la fibre optique) ou en ondes radio (comme les communications mobiles et Wi-Fi).

Ces connexions doivent être de qualité et éviter toute forme d'interférence pour supporter un service de qualité.

On connaît deux catégories de technologies de connexions :

- Les connexions LAN : au sein du réseau local.
- Les connexions WAN : qui interconnectent les sites distants.

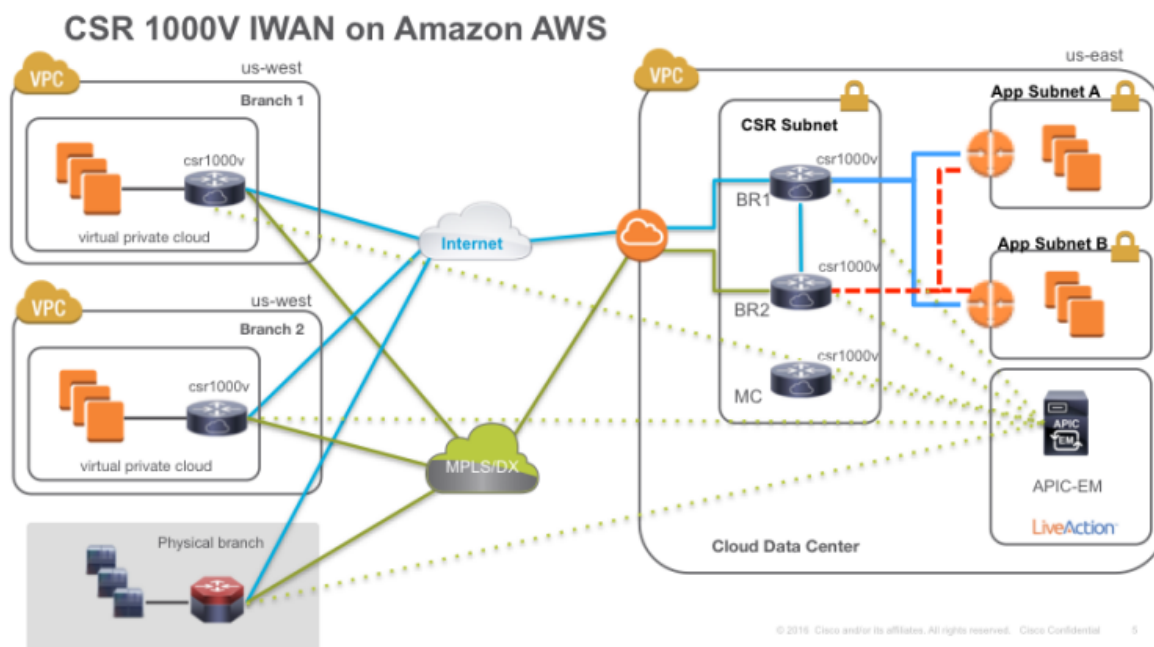
13. L'informatique en nuage

Un nuage : Un nuage représente une infrastructure dont on ne connaît pas vraiment la nature ou la topologie exacte et qui permet d'accéder à un réseau distant. Il s'agit typiquement d'un nuage Internet (au sens propre comme représentant un accès au réseau public) ou d'une simplification dans un diagramme. Le nuage est un diagramme courant dans nos topologies réseau bien avant l'apparition du paradigme des technologies en nuage. On pourrait aussi considérer un nuage comme la boîte noire de la cybernétique de Wiener.



Symbole d'un nuage

Mais justement avec l'informatique en nuage, du trafic d'entreprise pourrait utiliser des centres de données hébergés sous une autorité externe à celle-ci, c'est-à-dire dans le nuage au lieu d'utiliser de ressources sur site ("**On Premise**"). Les connexions par Internet entre les sites locaux et les services en nuage deviennent alors d'autant plus cruciales dans les nouvelles architectures.



Cisco CSR-1000v IWAN vers AWS

Dans cette topologie¹, on a plusieurs VPC (réseaux) dans le nuage AWS et un site physique. On peut transformer le réseau AWS en un IWAN “Hub and Spoke” en exécutant un CSR1000v comme BR (Broader Router). APIC-EM est hébergé dans le Hub (centre de données en nuage) pour fournir des services IWAN entre le hub IWAN et les sites distants, y compris virtuels et physiques.

Dans le cadre de ce modèle, certains services d’infrastructure se virtualisent jusqu’à être disponibles et utilisés en tant que services (*as a Service*) dont le réseau.

14. Cisco DNA

Cisco Digital Network Architecture (DNA) offre un centre de commande et de contrôle intuitif qui agit comme le cœur d’un réseau basé sur *l’intention* (IBN), fournissant un tableau de bord de gestion centralisé pour l’automatisation basée sur un contrôleur, l’assurance du réseau et l’extensibilité d’une plate-forme ouverte. Il utilise le “machine learning” et l’intelligence artificielle pour surveiller, dépanner et optimiser proactivement le réseau. Il permet d’utiliser des systèmes tiers pour améliorer les processus opérationnels.

DNA offre des fonctionnalités et des avantages en termes de gestion, d’automation, d’analytique et de sécurité.

- Gestion
 - Visibilité précise sur le réseau depuis un tableau de bord unique
 - Gestion du cycle de vie des périphériques
 - Intégration multidomaine
 - Ouverture et évolutivité
- Automation
 - Détection automatique des périphériques
 - Création de politiques par glisser-déposer
 - Gestion du déploiement et du cycle de vie des périphériques sans intervention
 - Qualité de service (QoS) automatisée

1. Source de l’image : [Cisco Blog Cisco SD-WAN Networking Service for Public Clouds](#)

- Analytique avec l'AI/ML
 - Utilisation de l'infrastructure en tant que capteur
 - Analytique contextuelle
 - Plus de 150 informations exploitables
 - Analytique avec l'intelligence artificielle locale et dans le cloud
- Sécurité
 - Détection et élimination des menaces
 - Intégration de Stealthwatch et ISE
 - Encrypted Traffic Analytics (ETA)
 - Segmentation ultra-sécurisée

DNA offre des solutions de type **Software Defined Network**. Le concept SDN est développé dans un chapitre ultérieur.

- [Network Assurance](#) : L'intelligence des réseau.
- [Software-Defined Access](#) : Le réseau d'accès agile.
- [Software-Defined WAN](#) : le réseau WAN agile.
- [Sécurité du réseau](#)

La solution est principalement construite autour de "DNA Center Appliance" et de périphériques réseaux ainsi que d'abonnements à des services.

Cisco DNA Center appliance



Cisco DNA Center appliance

Cisco DNA Center--running on the Cisco DNA Center appliance--is the network controller and command center for Cisco DNA. Use advanced artificial intelligence (AI) and machine learning (ML) to proactively monitor, troubleshoot, and optimize your network.

Cisco DNA infrastructure



Switching:
[Cisco Catalyst 9000 family switches](#)



Wireless access:
[Cisco Catalyst 9100 access points](#)



Wireless LAN controllers:
[Cisco Catalyst 9800 Series wireless controllers](#)



SD-WAN and routing:
[Cisco enterprise routers](#)

Your existing Cisco devices are probably compatible with this Cisco DNA solution.

Check the [list of Cisco DNA supported devices](#) to find out which Cisco DNA features and capabilities your existing Cisco devices support and to make sure those devices are running the latest Cisco iOX software.

Cisco DNA Center Appliance et ses périphériques

15. Éléments clés à retenir

Il est essentiel de pouvoir identifier le rôle, les fonctions, les formats, la portée protocolaire et d'architecture des différents périphériques du réseau.

4. Topologies du réseau

Dès que l'on est en mesure d'identifier les composants du réseau et leur rôle fondamental, on peut concevoir des topologies du réseau. Ce chapitre vise à présenter différentes topologies conceptuelles en étoile, maillée, point-à-point dans le cadre des technologies étudiées dans le CCNA.

Une topologie indique les chemins possibles que les informations peuvent prendre lors d'une communication d'un hôte terminal d'origine à un hôte terminal de destination. Les différents périphériques intermédiaires remplissent un fonction de transfert de couche 2 (L2, commutation *frame switching*) et/ou de couche 3 (L3, routage IP, *IP Routing*, *packet forwarding*), voire de filtrage notamment à des fins de sécurité (IPS, IDS, pare-feu, *firewall*) ou de contrôle (APIC-EM, Cisco DNA), ...

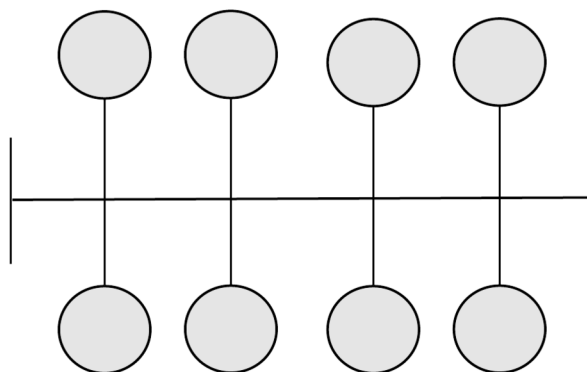
Un présentation des modèles de conception, dans une vue plus logique et d'optimisation de ces trajets faisant appel aux modèles "2 tier", "3 tier" et "Leaf-Spine" est développée dans le chapitre "[Principes de conception LAN](#)".

1. Topologies conceptuelles

Dans ces topologies conceptuelles de base, on sera attentif au nombre de liaisons créées entre les points. Si des chemins uniques ou des points uniques sont des points de rupture, des chemins multiples compliquent les calculs des meilleurs trajets d'un point à un autre !

1.1. Topologies simples en bus, point à point et en anneau

Dans une topologie en bus, un seul canal est partagé par des points multiples. Une rupture de connexion empêcherait tout point de communiquer avec l'autre.

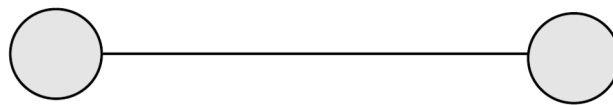


Topologie en bus

Topologie en bus

Dans une topologie point-à-point, la seule destination est l'autre point. La ligne est un point unique de défaillance (*single point of failure*, soit un point d'un système informatique dont le reste du système est dépendant et dont une panne entraîne l'arrêt complet du système.)¹.

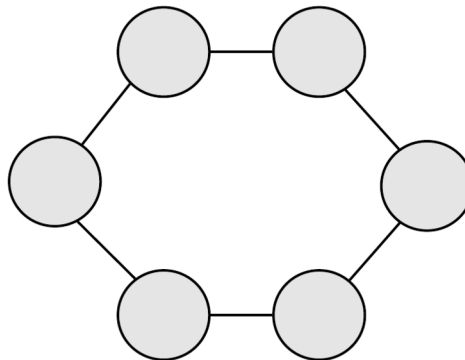
1. https://fr.wikipedia.org/wiki/Point_individuel_de_d%C3%A9faillance



Topologie point-à-point

Topologie point-à-point

Dans une topologie en anneau, la rupture d'une liaison n'empêchera aucun point de communiquer avec l'autre.

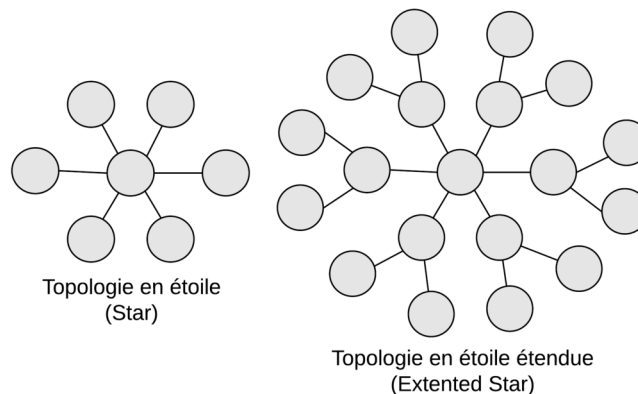


Topologie en anneau

Topologie en anneau

1.2. Topologie en étoile (star) et en étoile étendue

Dans une simple topologie en étoile, le point de concentration est un point unique de défaillance. Selon l'endroit d'une rupture dans une topologie en étoile étendue, une partie des noeuds seront isolés les uns par rapport à autres noeuds.

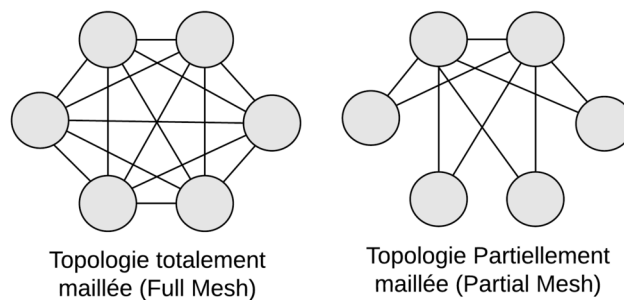


Topologie en étoile (star) et en étoile étendue

1.3. Topologie maillée (total mesh) et Topologie hybride (partial mesh)

Dans une topologie maillée, tous les noeuds sont interconnectés entre eux de telle sorte que les communications peuvent résister à des ruptures multiples. La redondance est maximale. Chaque noeud ajouté augmente de manière exponentielle le nombre de liaisons pour assurer le maillage.

$$\text{Connexions d'une topologie maillée} = \frac{n \times (n - 1)}{2}$$



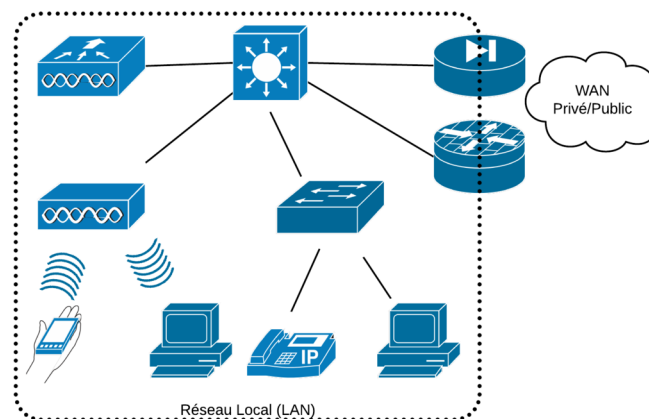
Topologie maillée (total mesh) et Topologie hybride (partial mesh)

Dans une topologie partiellement maillée, chaque noeud d'extrémité se connecte à $n + 1$ points de concentration. Une topologie partiellement maillée est une topologie commune dans les architectures du réseau.

2. Topologies de base

2.1. Un LAN d'entreprise

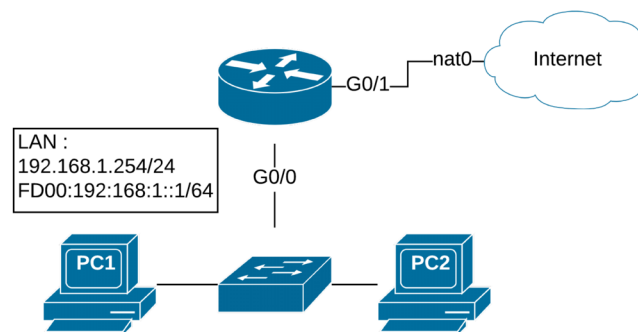
Dans la figure suivante, le réseau local (LAN) correspond à une topologie en étoile.



Un réseau local LAN simple

2.2. SOHO (Small Office / Home Office)

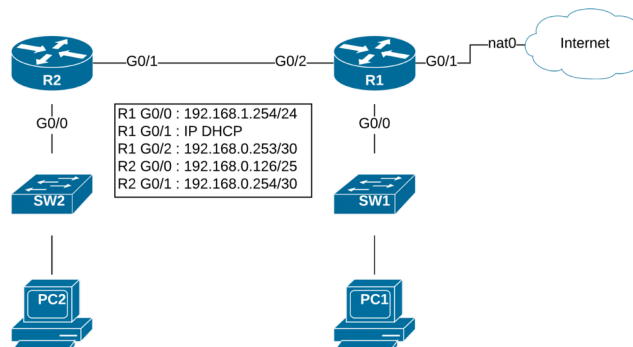
Un réseau local en topologie en étoile se connecte à un Internet (nuage) typiquement un petit bureau, voire à la maison (Small Office / Home Office, "SoHo")



Un simple LAN

2.3. Deux sites interconnectés avec un Internet

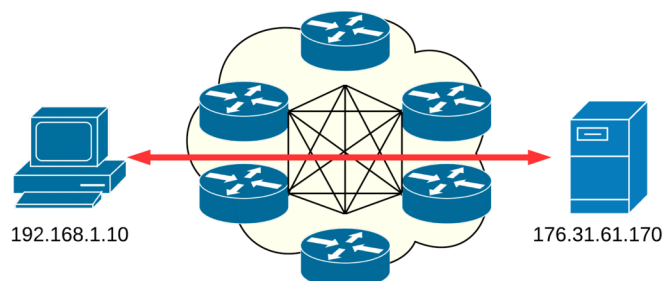
Deux sites sont interconnectés en “point-à-point” par exemple avec la connexion d’un fournisseur de service de type ligne louée (qu’importe la technologie physique utilisée). *A priori*, les LANs des deux site sont en étoile. L’un des sites connectent l’Internet et offre cette connectivité au second site (à gauche dans la figure).



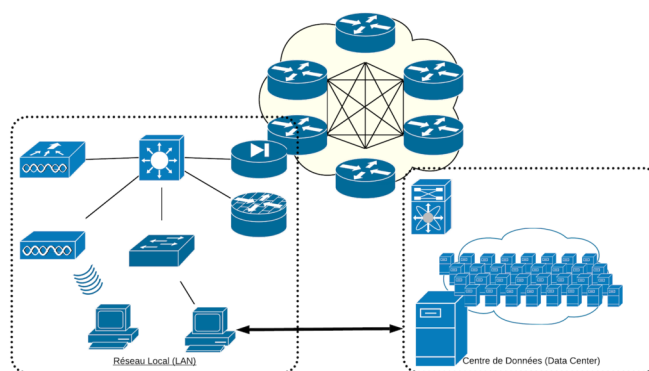
Deux sites et un Internet

2.4. Un Internet maillé

On trouve dans les deux figures suivantes un Internet totalement maillé (à titre conceptuel). Par contre, les images suggèrent qu’à travers ce maillage, en surcroupe, les deux hôtes d’extrémités se connectent en “point-à-point”.



Un Internet maillé



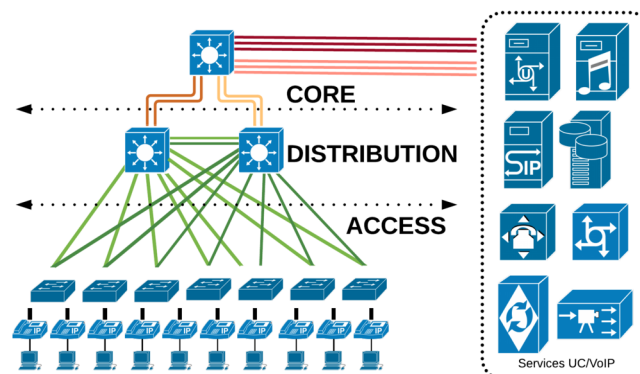
Une connexion TCP entre un LAN et un serveur dans l’Internet

3. Topologies LAN

3.1. Modèle de conception Cisco Systems

Les modèles de conception de réseau sont exposés dans le document “[LAN Design, Conception LAN](#)”.

Les périphériques sont répartis par couches fonctionnelles “Core”, “Distribution”, “Access” qui s’interconnectent entre elles selon certaines règles de conception pour assurer la Haute Disponibilité de l’infrastructure.

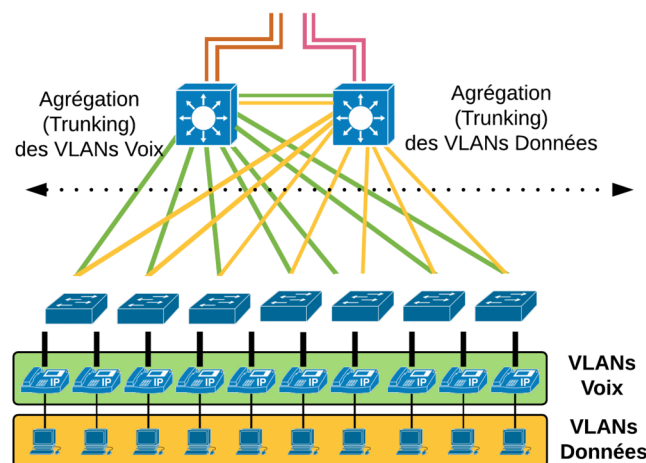


Couches “Core”, “Distribution”, “Access”

Dans ce genre de topologie, une hiérarchie est mise en place avec du maillage.

3.2. Conception LAN avec VLANs

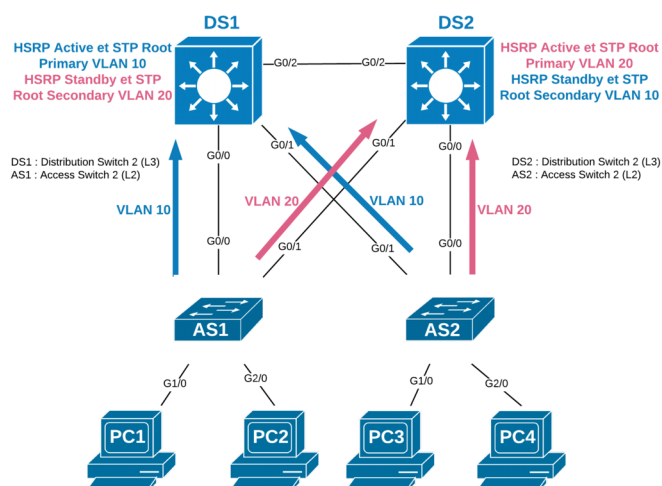
Entre la couche “Access” et la couche “Distribution”, on trouve un maillage partiel. Chaque commutateur de la couche “Access” on une connexion avec deux commutateurs.



Conception LAN avec VLANs

3.3. Topologie Campus LAN

Voici une topologie intitulée “Load Balancing Per-VLAN Rapid Spanning-Tree”, soit répartition de charge par VLANs avec Rapid Spanning-Tree et avec HSRP.



Topologie Campus LAN, Load Balancing Per-VLAN Rapid Spanning-Tree et HSRP

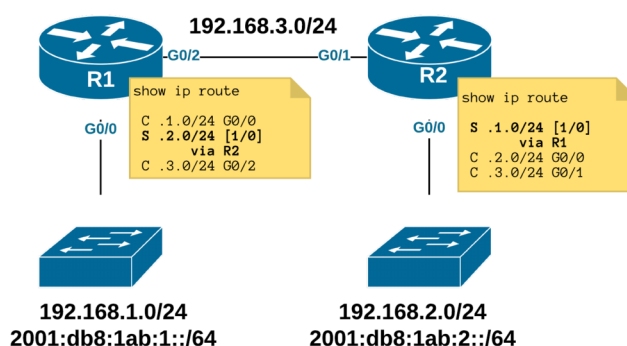
Dans cette topologie, chaque commutateur d'accès (AS1 et AS2) place le trafic d'un VLAN sur un seul et même commutateur de distribution (VLAN 10 pour DS1 et VLAN 20 pour DS2) sur des liaisons distinctes. Par défaut, sans répartition de charge des VLANs sur des liaisons distinctes, l'une d'elles serait nécessairement bloquée et l'autre embarquerait le trafic de tous les VLANs. Au cas où une liaison entre les deux couches venait à tomber, l'autre reprendrait immédiatement en charge le trafic de l'autre VLAN.

Ensuite, dans cette topologie chacun des deux commutateurs est passerelle par défaut pour son VLAN principal et passerelle de sauvegarde pour l'autre VLAN. De la sorte, on assure la disponibilité des passerelles par défaut et un accès vers d'autres réseaux.

Enfin, on remarquera que la connectivité de la couche "Access" ne dépend d'aucun autre commutateur de même couche.

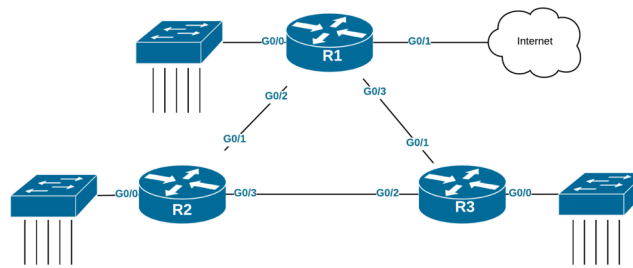
4. Topologies WAN basiques

4.1. Connexion point à point sur une ligne dédiée entre deux sites



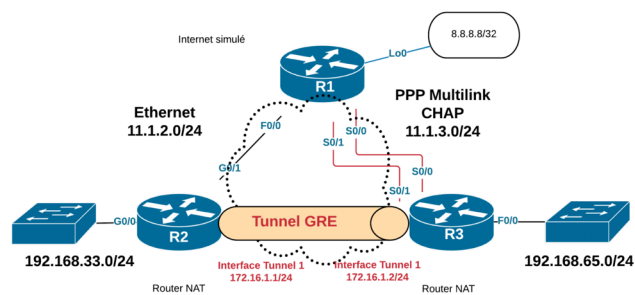
Connexion point à point sur une ligne dédiée entre deux sites

4.2. Un maillage WAN de trois routeurs



Un maillage WAN de trois routeurs

4.3. Topologie WAN tunnel GRE



Topologie WAN tunnel GRE

5. Mathématiques des réseaux

1. Introduction

Les conversions de valeurs en binaire, en décimal et en hexadécimal sont des fondamentaux utiles pour la manipulation des identifiants réseaux tels que des adresses IP ou des adresses MAC. Aussi, le trafic est codé sur le support en binaire mais les décodeurs et analyseurs de paquets offrent une vue du trafic en hexadécimal. Il s'agit ici de se familiariser avec ces méthodes de représentation.

1.1. Objectifs

- Conversions et manipulations binaires, décimales et hexadécimales.
- La manipulation des adresses MAC-48 MAC-EUI64.
- IPv4 et IPv6 demandent que l'on s'intéresse aux différents systèmes de numération impliqués :
 - binaire,
 - décimal,
 - hexadécimal, voire octal.

1.2. Systèmes de numération

Pour coder des adresses, on peut utiliser plusieurs modes de numérations :

- en base 10 (décimal) : un symbole à dix valeurs, notre habitude ;
- en base 2 (binaire) : un symbole à deux valeurs, facile mais fastidieux ;
- en base 16 (hexadécimal) : un symbole à seize valeurs, efficace compte tenu des besoins actuels.

1.3. Représentation

- En décimal, on utilise 0 à 9 (10 valeurs).
- En binaire, on utilise soit 0, soit 1 (deux valeurs).
- En hexadécimal, 0x0 à 0x9, 0xA, 0xB, 0xC, 0xD, 0xE et 0xF (16 valeurs).

La simplification de la représentation des valeurs est intéressante ; par exemple : 63789 en décimal est représenté par 1111 1001 0010 1101 en binaire et par 0xF92D en hexadécimal.

1.4. Unités de mesure

Les ordinateurs ou les processeurs modernes utilisent généralement des blocs de données de 8, 16, 32 ou 64 bits bien que d'autres tailles soient aussi possibles. La nomenclature actuelle est comme suit :

- donnée de 8 bits : "octet", "byte" ;
- donnée de 16 bits : "word" ou "mot" ;
- donnée de 32 bits : "dword" ou "double mot" ;
- donnée de 64 bits : "qword" ou "quadruple mot".

1.5. Codage des adresses réseau

Adresses MAC : 48 bits représentés en douze hexas organisés en 6 octets ou en 3 mots. Par exemple :

00:d1:81:41:d2:00

Adresse IPv4 et son masque : 32 bits organisés en 4 octets représentés par des décimales séparées par des points. Par exemple :

192.168.3.1

Adresses IPv6 : 128 bits organisés en 8 mots de 16 bits représentés en hexas séparés par des deux-points. Par exemple :

fe80::2d1:81ff:fe41:d200

2. Adresse IPv4

2.1. codage et étendue

Une adresse IPv4 connaît 2^{32} possibilités soit en décimal 4 294 967 296 possibilités.

Une adresse IPv4 est codée en quatre octets représentés en notation décimale (pointée), les valeurs de chaque octet pouvant varier entre 0 et 255 et sont séparées par un point ..

[0-255] . [0-255] . [0-255] . [0-255]

En ce qui concerne nos besoins dans la compréhension des réseaux, si on se contente des valeurs à la limite des octets, il n'est pas nécessaire d'être capable de convertir du décimal vers le binaire, mais comme ce ne sera pas toujours le cas ...

2.2. Conversion décimale/binaire

Pour réaliser une conversion décimale vers binaire, on crée un tableau de huit valeurs à la puissance deux.

Par exemple pour convertir 125 en un octet en binaire :

Combien de fois 128 dans 125 ?

- → 0 fois

Combien de fois 64 dans 125 ?

- → 1 fois, reste 125-64=61

Combien de fois 32 dans 61 ?

- → 1 fois, reste 61-32=29 Combien de fois 16 dans 29 ?
- → 1 fois, reste 29-16=13

Combien de fois 8 dans 13 ?

- → 1 fois, reste 13-8=5

Combien de fois 4 dans 5 ?

- $\rightarrow$ 1 fois, reste $5-4=1$

Combien de fois 2 dans 1 ?

- $\rightarrow$ 0 fois,

Combien de fois 1 dans 1 ?

- $\rightarrow$ 1 fois, reste $1-1=0$

.	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
125	0	1	1	1	1	1	0	1

2.3. Conversion binaire vers décimal

L'octet codé 01111101 est porté dans un tableau de puissance de 2

.	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
125	0	1	1	1	1	1	0	1

On additionne $64+32+16+8+4+1 = 125$

2.4. Valeurs possibles d'un octet dans un masque

On trouvera ici les valeurs possibles de un octet dans un masque d'adresse IPv4.

Valeur décimale	Valeur binaire	Bits à 1	Bits à 0
0	00000000	0	8
128	10000000	1	7
192	11000000	2	6
224	11100000	3	5
240	11110000	4	4
248	11111000	5	3
252	11111100	6	2
254	11111110	7	1
255	11111111	8	0

2.5. Opération binaire ET

Pour le protocole IPv4 il est utile de calculer rapidement le résultat de l'opération binaire ET :

```

      1 1 0 0
ET  1 0 1 0
-----
      1 0 0 0

```

2.6. Exemple d'opérations binaires ET

Par exemple, 125 ET 252 en binaire :

Opération	Décimale	Binaire
-	125	01111101
ET		
-	252	11111100
donne		
-	124	01111100

2.7. Multiple

Un **multiple** de n est le produit de n par un nombre entier. En bref, c'est le **résultat d'une multiplication de n** .

Quel est multiple de 8 inférieur ou égal à 90 ? c'est 88.

2.8. Logarithme binaire (en base 2)

En mathématiques, le logarithme binaire ($\log_2 n$) est le logarithme de base 2. C'est la fonction réciproque de la fonction puissance de deux : $x \mapsto 2^x$ ¹.

Quel est le logarithme binaire de 64 ? C'est 6 car 2^6 est égal à 64.

Quel est le logarithme binaire de 8 ? C'est 3 car 2^3 est égal à 8.

2.9. Conversion d'un masque de réseau

Pour convertir un masque de réseau IPv4 d'une notation décimale pointée vers une notation CIDR.

La notation décimale pointée exprime le masque de réseau en 8 bits décimaux séparés par des points.

La notation CIDR indique le nombre de bits à 1 dans le masque.

Si nous savons qu'un masque de réseau est nécessairement une suite de bits à 1 et puis seulement de zéro.

Si nous savons que huit bits à un correspondent à la valeur décimale 255 et si nous savons que huit bits à zéro correspondent à la valeur décimale 0.

Alors, il nous suffit de retenir le tableau des suites de bits à un et puis à zéro sur un seul octet (huit bits) pour maîtriser les valeurs d'un masque de réseau :

Bits	Binaire	Décimal
0	00000000	0
1	10000000	128
2	11000000	192
3	11100000	224
4	11110000	240
5	11111000	248
6	11111100	252
7	11111110	254
8	11111111	255

3. Adresses MAC IEEE 802

3.1. Adresses MAC-48 IEEE 802

Il n'y a pas de calcul sur les adresses MAC IEEE 802.3 à maîtriser.

Les adresses MAC sont constituées de douze chiffres représentés en hexadécimal ($12 \times 4 \text{ bits} = 48 \text{ bits}$), souvent groupées en octets (deux hexas) :

1. https://fr.wikipedia.org/wiki/Logarithme_binaire

00:d1:81:41:d2:00

Où les 24 premiers bits identifient le constructeur de la carte (Organizationally Unique Identifier - OUI) et où les 24 derniers bits sont laissés à la discrétion des fabricants de cartes réseau (NIC, Network Interface Card).

3.2. Adresses MAC EUI 64

On étend une adresse MAC-48 00:d1:81:41:d2:00 en MAC-EUI64 :

1) en découpant l'adresse MAC en deux parties égales de 24 bits en son milieu :

00:d1:81:--:--:41:d2:00

2) en y insérant 16 bits ff : fe :

00:d1:81:ff:fe:41:d2:00

3) Il est nécessaire d'inverser le bit "Universal/Local" ("U/L bit") à la septième position du premier octet. Le bit "u" est fixé à 1 (Universal) et il est fixé à zéro (0) pour indiquer une portée locale.

02:d1:81:ff:fe:41:d2:00

La représentation donne ceci en trois groupes de 16 bits :

02d1:81ff:fe41:d200

4. Adresses IPv6

4.1. Blocs IPv6 /64

Les adresses IPv6 sont constituées de mots de 16 bits notés en hexadécimal.

Par exemple :

```
fd00:2001:0db8:0002:02d1:81ff:fe41:d200/64
-----
16b. 16b. 16b. 16b. 16b. 16b. 16b. 16b. masque
```

Les 64 premiers bits déterminent le préfixe IPv6, soit :

fd00:2001:0db8:0002::/64

Les 64 derniers bits déterminent l'identifiant d'interface, soit :

02d1:81ff:fe41:d200

4.2. Blocs IPv6 /48

Dans les découpages de blocs d'entreprise on trouvera un /48. Par exemple :

fd00:2001:0db8::/48

Il reste donc 16 bits pour créer 2^{16} (65 536) sous-réseaux /64 :

- fd00:2001:0db8:0000::/64, fd00:2001:0db8:0001::/64
- fd00:2001:0db8:0002::/64, fd00:2001:0db8:0003::/64, ...
- fd00:2001:0db8:00fe ::/64, fd00:2001:0db8:00ff ::/64
- fd00:2001:0db8:0010::/64, fd00:2001:0db8:0011::/64, ...
- fd00:2001:0db8:fffc ::/64, fd00:2001:0db8:fffd ::/64
- fd00:2001:0db8:fffe ::/64, fd00:2001:0db8:ffff ::/64

4.3. Conversion binaire / hexadécimal / decimal

Hexadécimal	Binaire	Décimal
0x0	0000	0
0x1	0001	1
0x2	0010	2
0x3	0011	3
0x4	0100	4
0x5	0101	5
0x6	0110	6
0x7	0111	7
0x8	1000	8
0x9	1001	9
0xA	1010	10
0xB	1011	11
0xC	1100	12
0xD	1101	13
0xE	1110	14
0xF	1111	15

5. Éléments clés à retenir

- Il est utile de savoir convertir 8 bits en décimal, binaire et hexadécimal.
- Il est utile de connaître les valeurs sur un octet d'une suite de bits à 1 et puis de bits à 0.
- On manipule utilement des blocs de 128, 64, 48, 24 et 16 bits en hexadécimal.

Deuxième partie Cisco IOS CLI

Cette partie évoque le système d'exploitation Cisco IOS, la solution Open Source de simulation d'infrastructures informatiques [GNS3](#). On se demandera malgré tout comment accéder à l'IOS avec une console physique, comment reprendre la main sur les routeurs et les commutateurs Cisco avec un “password recovery”, avec une initiation à la méthode de configuration en ligne de commande IOS CLI.

6. Cisco IOS Internetwork Operating System

1. Introduction

Cisco IOS (Internetwork Operating System) est une famille de logiciels utilisée sur la plupart des routeurs et commutateurs Cisco Systems. IOS dispose d'un ensemble de fonctions de routage (*routing*), de commutation (*switching*), d'interconnexion de réseaux (*internetworking*) et de télécommunications dans un système d'exploitation multitâche.

La plupart des fonctionnalités IOS ont été portées sur d'autres noyaux comme QNX (IOS-XR) et Linux (IOS-XE).

Tous les produits Cisco ne fonctionnent pas nécessairement sous Cisco IOS, comme les plateformes de sécurité ASA (dérivé Linux) ou les routeurs "carrier" qui fonctionnent sous Cisco IOS-XR.

2. Versions

La dénomination des versions Cisco IOS utilise des nombres et certaines lettres, en général sous la forme a.b(c.d)e où :

- a est le numéro de version **majeur**
- b est le numéro de version **mineur**
- c est le numéro de révision (release)
- d est le numéro de l'interim (omit des révisions générales)
- e (aucune, une ou deux lettres) est l'identifiant du train comme aucun (Mainline), T (Technology), E (Enterprise), S (Service provider), XA est un train de fonctionnalités spécifiques, etc.

Rebuilds – Un rebuild est souvent une version corrective compilée d'un problème ou d'une vulnérabilité pour une version IOS donnée. Par exemple, 12.1(8)E14 est un Rebuild, le 14 signifiant le 14ème rebuild de 12.1(8)E.

Interim releases – Basé sur une production hebdomadaire.

Maintenance releases – Révision rigoureusement testées.

3. Trains

Un train est un véhicule fournissant un logiciel Cisco auprès d'un ensemble de plateformes et de fonctionnalités.

3.1. Jusqu'en version 12.4

Avant l'IOS release 15, les révisions étaient séparées en différents "trains", chacun contenant un ensemble de fonctionnalités. Les "trains" étant liés aux différents marchés et clients que Cisco voulait toucher.

- **The mainline train**, Version stable.
- **The T – Technology train**, Version en développement, prochaine version stable.
- **The S – Service Provider train**
- **The E – Enterprise train**
- **The B – broadband train**
- **The X* (XA, XB, etc.)**

3.2. Depuis 15.0

À partir de la version IOS 15, il n'y a plus qu'un seul train : le Train M/T.

4. Packaging / feature sets

La plupart des produits qui fonctionnent en Cisco IOS ont une ou plusieurs "feature sets" ou "packages", 8 pour les routeurs Cisco et 5 pour les switches Cisco.

Par exemple, pour un switch Catalyst les IOS sont disponibles en versions :

- "standard" : Routage IP de base
- "enhanced" : Full routage IPv4
- "advanced IP services" : Full routage IPv6

On peut trouver une fonctionnalité grâce au "Cisco Feature Set Browser" ([Cisco Feature Navigator](#)).

Avec les routeurs ISR 1900, 2900 and 3900, Cisco a révisé son modèle de licence. Les routeurs viennent avec une licence de base et celle-ci est étendue avec une fonctionnalité déjà native au logiciel par activation.

- **Data** : BFD, IP SLAs, IPX, L2TPv3, Mobile IP, MPLS, SCTP.
- **Security** : VPN, Firewall, IP SLAs, NAC.
- **Unified Comms** : CallManager Express, Gatekeeper, H.323, IP SLAs, MGCP, SIP, VoIP, CUBE(SBC).

On trouvera plus d'information sur les versions Cisco IOS dans l'article publié chez Cisco Press "[An Overview of Cisco IOS Versions and Naming](#)".

5. Cisco IOS

Le système d'exploitation Cisco IOS a été développé depuis les années 1980 pour des routeurs disposant de faibles ressources en RAM (256 Kb) et en CPU. La modularité dans l'architecture Cisco IOS a permis sa croissance sur du matériel toujours mieux adapté en fonctionnalités nouvelles.

Dans toutes les versions de Cisco IOS, le routage de paquets (*packet routing*) et la commutation de paquets (*packet switching*) sont des fonctions distinctes.

Le routage et les autres protocoles fonctionnent en tant que processus IOS et contribuent à la table "Routing Information Base (RIB)". Celle-ci subit un traitement pour produire la table finale "IP forwarding table (FIB, Forwarding Information Base)". La FIB est utilisée par la fonction de transfert (*forwarding*) du routeur.

Cisco IOS est construit selon une architecture monolithique. Il fonctionne comme une seule image et tous les processus partagent le même espace mémoire. Il n'y a aucun mécanisme de protection entre les processus. En d'autres termes, un bug dans l'IOS peut potentiellement corrompre des données utilisées par un autre processus. Le noyau de l'IOS n'est pas préemptif.

Type	noyau	Mémoire
IOS	Monolithique, non-préemptif	mémoire partagée sans protection entre processus, pas de "swapping" ou "paging", plus de performance moins de sécurité

Plateformes fonctionnant en Cisco IOS monolithique.

Gamme de plateformes	Type de plateformes
Routeurs ISR G1 et ISR G2	Cisco 800, 1800, 1900, 2800, 3200, 3800, 2900, 3900, 3600 et 3700.
Commutateurs Cisco Catalyst	2970, 3560 3750, 4500, 4900 et 6500.
Routeur virtuel générique de lab	IOSv

6. Cisco IOS-XR

Pour les produits Cisco nécessitant de la haute disponibilité comme les [Cisco CRS](#), les limites d'un IOS monolithique ne sont plus acceptables d'autant plus qu'un système d'exploitation d'un compétiteur comme JUNOS de Juniper de 10 à 20 ans son cadet ne connaît pas ces limites grâce à sa base UNIX.

Une réponse de Cisco a été de concevoir une nouvelle version de Cisco IOS appelée Cisco IOS-XR (2006) offrant plus de modularité et une protection entre les processus, des "lightweight threads", de l'ordonnancement préemptif et la capacité de redémarrer indépendamment des processus qui ont échoué. IOS-XR utilise un micro-noyau tiers en temps réel ("a 3rd party real-time operating system microkernel") appelé QNX. Une bonne partie du code source de l'IOS a été réécrit pour profiter du nouveau noyau. L'avantage d'un micro-noyau est qu'il élimine tout processus qui n'est absolument pas nécessaire d'une part, et d'autre part qu'il les exécute ces processus comme des processus d'application.

Grâce à cette méthode, IOS-XR est capable d'atteindre le niveau de haute disponibilité que doivent atteindre les nouvelles plateformes. Même si les systèmes sont de conception franchement différente, ils sont très proches quant à leur usage.

Type	noyau	Mémoire
IOS-XR	third-party real-time operating system (RTOS) microkernel, QNX, multitâches préemptif, protection entre les processus, des "lightweight threads", de l'ordonnancement préemptif et la capacité de redémarrer indépendamment des processus qui ont échoué	mémoire dédiée et protégée.

Plateformes IOS-XR :

- Cisco CRS-1 Carrier Routing System (CRS),
- CRS-3 Carrier Routing System,
- ASR9000 Series routers,
- Cisco XR 12000 Series routers pour service provider core networks,
- IOS-XRv



Cisco CRS 16-Slot Back-to-Back (2+0) System

Source de l'image : [Cisco CRS 16-Slot Back-to-Back \(2+0\) System](#)

7. Cisco IOS-XE

L'IOS-XE utilise des composants d'architecture qui l'améliore et le différencie par rapport à un noyau IOS traditionnel. Toutefois, la ligne de commande Cisco IOS CLI (Command Line Interface) et les procédures de configuration sont très proches de telle sorte que les opérateurs puissent passer de l'un à l'autre facilement. Il fonctionne sur des machines physiques dédiées ou virtuelles avec des processeurs Intel 64 bits.

IOS-XE ajoute la stabilité grâce à l'abstraction de la plateforme supportée par un **noyau Linux**. Le noyau Linux et ses pilotes sont les seuls composants de l'architecture IOS-XE qui disposent d'un accès direct au matériel en fournissant une stabilité supplémentaire.

Le noyau Linux permet d'exécuter des processus sur de multiples processeurs. Les bibliothèques logicielles permettent à des applications de fonctionner sur le même matériel. L'exécution de Wireshark sur un Superviseur Catalyst 4500 est un exemple.

Tous les protocoles de routage sont des démons appelés *IOSd* qui fonctionnent comme des processus distincts.

IOS-XE supporte des architectures Intel 64 bits permettant une exploitation de la mémoire vive au-delà de 4 Go (limite associée aux architectures 32 bits). Le noyau attribue de la mémoire vive aux processus IOSd. Les IOSd attribuent alors une taille de mémoire configurable et les utilisent pour diverses fonctions de l'IOS.

Plateformes IOS-XE

- [ASR1000](#)
- [CSR1000v](#)
- Cisco ASR 900 Series
- Catalyst 4000 Series Supervisor 7



Gamme Cisco ASR1000

Source de l'image : [Gamme Cisco ASR1000](#)

Cisco Systems et le programme de la certification CCNA nous invitent à nous intéresser à la programmation des infrastructures réseau à partir de Cisco IOS-XE comme par exemple sur <https://developer.cisco.com/site/ios-xe/>.

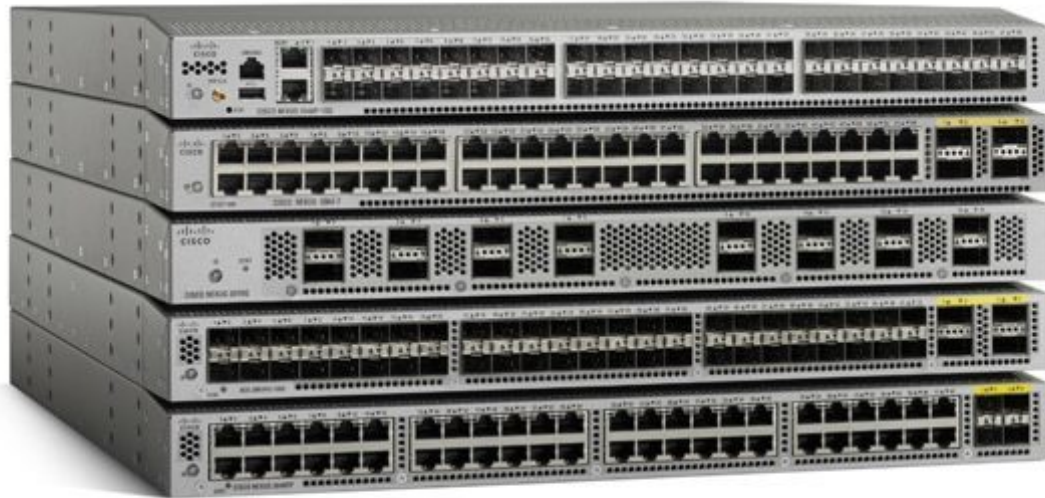
8. Cisco NX-OS

Cisco a fabriqué un système d'exploitation de prochaine génération pour le *data center* pour un maximum d'évolutivité et de disponibilité des applications. NX-OS est une évolution du système d'exploitation industriel Cisco Storage Area Network Operating System (SAN-OS) Software et fonctionne avec une base Wind River Linux¹.

Plateformes IOS NX :

- [Cisco Nexus 3000](#),
- [Cisco Nexus 5000](#),
- [Cisco Nexus 7000](#),
- [Cisco Nexus 9000](#),
- Cisco 9k series routers,
- NX-OSv

1. https://en.wikipedia.org/wiki/Cisco_NX-OS

*Cisco Nexus 3000*

Source de l'image : [Cisco Nexus 3000](#)

9. Cisco ASA OS

La gamme “Adaptive Security Appliance” (ASA) est celle des pare-feu chez Cisco. [Cisco ASA Software](#) est le système d'exploitation développé sur ces plateformes. Il fonctionnait sous d'anciennes plateformes comme Cisco PIX, Cisco VPN 3000, Cisco IPS 4200. L'interface Cisco AS-OS est très proche de celle des plateformes IOS.

- <http://www.cisco.com/c/en/us/products/security/adaptive-security-appliance-asa-software/index.html>
- <http://www.cisco.com/c/en/us/products/security/virtual-adaptive-security-appliance-firewall/index.html>
- <https://docs.gns3.com/appliances/cisco-asav.html>

*Gamme Cisco ASA*

Source de l'image : [Gamme Cisco ASA](#)

10. Cisco IOx

L'environnement applicatif Cisco® IOx combine l'exécution d'applications IoT dans le brouillard, une connectivité sécurisée avec le logiciel Cisco IOS® et des services puissants pour une intégration rapide et fiable avec les capteurs de l'Internet des objets (IoT) et le nuage. En apportant la capacité d'exécution d'applications à la source des données IoT, les clients surmontent les difficultés liées aux volumes élevés de données et à la nécessité d'une réactivité du système automatisée en temps quasi réel. Cisco IOx offre une gestion et un hébergement cohérents pour tous les produits d'infrastructure réseau, y compris les routeurs, les commutateurs et les modules de calcul Cisco. Cisco IOx permet aux développeurs d'applications de travailler dans l'environnement familier des applications Linux avec leur choix de langages et de modèles de programmation avec des outils de développement open-source familiers.²

Cisco IOx peut être activé sur les plateformes suivantes :

- Cisco 800 Series Industrial Integrated Services Routers
- Cisco Industrial Ethernet 4000 family of switches
- Compute Module for Cisco 1000 Series Connected Grid Routers
- Cisco IR510 WPAN Industrial Router

En termes d'architecture matérielle, à titre d'exemples, les routeurs matériels IR829 et IR809 utilisent le même processeur Intel Rangeley double cœur, avec 2 Go de mémoire DDR3, 8 Mo de mémoire SPI Bootflash et 8 Go (4 Go utilisables) de mémoire flash eMMC.³

2. [Cisco IOx Data Sheet](#).

3. [Documentation>IOx>IR8xx Platforms](#)

L'hyperviseur, fourni par LynxWorks, fonctionne sur le matériel baremetal sur lequel l'IOS et un OS invité (par exemple Linux) fonctionnent comme deux machines virtuelles (VM) séparées.

L'hyperviseur présente le matériel sous-jacent aux machines virtuelles (IOS et OS invité) comme un sous-ensemble du matériel physique réel – ce qui permet la virtualisation imbriquée (*Nested virtualization*). La configuration de l'hyperviseur permet de déterminer quels périphériques doivent être affectés à quelle VM. Les VM accèdent à une unité centrale virtuelle, à des zones de mémoire préconfigurées, à un stockage sur disque flash prédivisé et à d'autres périphériques matériels. LynxSecure Separation Kernel v. 5.1 a été sélectionné comme hyperviseur.

Chaque périphérique PCI peut être détenu exclusivement par une VM dans l'architecture de l'hyperviseur. Toutefois, l'IOS et le système d'exploitation invité doivent accéder à certains périphériques partagés, par exemple la mémoire flash eMMC. La solution est un serveur de périphériques virtuels (VDS), qui est une VM distincte possédant des périphériques partagés. L'IOS et le système d'exploitation invité accèdent aux dispositifs virtuels émulés dans l'hyperviseur. L'hyperviseur et le VDS coordonnent ensuite l'accès aux dispositifs partagés. Le VDS fournit également des canaux de communication entre les VM, en utilisant des interfaces Ethernet émulées.

L'IOS agit comme une passerelle vers les ressources réseau pour la partition Linux, en utilisant l'adresse IP et une connexion de commutateur virtuel fournie à l'hyperviseur. IOS et Linux fonctionnent chacun comme un système d'exploitation invité de l'hyperviseur.

Le chemin réseau de IOXVM à IOS est disponible via une liaison Ethernet émulée entre eux. Le protocole IPv4 ou IPv6 peut être exécuté sur la liaison Ethernet.

L'IOXVM exécute le CAF et d'autres éléments de l'infrastructure IOx sur le Linux hôte et héberge toutes les applications LXC.

On imagine donc que des machines virtuelles Qemu/KVM ou des containers Docker puissent être intégrés au plus proche des clients dans le matériel réseau.

Vous pouvez commencer avec Cisco IOx à partir de la page [“What is IOx ?”](#)

11. Sources du document

- Pearson VUE, [An Overview of Cisco IOS Versions and Naming](#)
- [Cisco CRS](#)
- <https://developer.cisco.com/site/ios-xe/>
- https://en.wikipedia.org/wiki/Cisco_ASA
- [“What is IOx ?”](#)

7. Installer et configurer GNS3

Pour se rapprocher d'une expérience moderne des interfaces de configuration des produits Cisco Systems, on conseillera volontiers le projet Open Source GNS3 comme meilleur rapport qualité/prix pour se préparer aux examens de certification et à la pratique des systèmes Cisco. Voici une guide d'installation et de configuration.

Au sommaire de ce document, on développera les sujets suivants :

1. Introduction à GNS3
2. Architecture Client / Serveur
3. Installation de GNS3 Server chez Scaleway
4. Comment trouver des images Cisco IOS pour GNS3 ?
5. Installation de GNS3 GUI et connexion au serveur
6. Appliances GNS3
7. Création et manipulation d'un projet GNS3
8. Sujets avancés

1. Le logiciel GNS3

The software
that empowers
network
professionals.

<https://gns3.com/>



Logo de GNS3

1.1. Présentation de GNS3

GNS3, "Graphical Network Simulator-3", est une interface graphique frontale et une plateforme de contrôle écrites en Python pour simuler des infrastructures informatiques. Les technologies de virtualisation utilisées par GNS3 sont Dynamips, VPCS, VMWare Workstation/ESXi, VirtualBox, QEMU/KVM, Docker, ...

Le logiciel émule aussi des technologies LAN/WAN comme Ethernet, Frame-Relay, ATM, HDLC, ... Il permet d'interconnecter les ordinateurs émulés sur des commutateurs virtuels ou des interfaces physiques du serveur.

Une topologie peut être distribuée sur plusieurs serveurs. Il permet de construire des topologies complexes.

Il est possible de capturer en temps réel le trafic qui passe par les interfaces des périphériques.

Les topologies créées (on parle de *projets GNS3*) sont importables/exportables. Pour se connecter aux périphériques d'une topologie, on utilise une console texte ou une console graphique.

On trouvera les fonctionnalités complètes sous ces liens :

- [What's new in GNS3 version 2.0](#)
- [What's new in GNS3 version 2.1](#)
- [What's new in GNS3 version 2.2](#)
- [GNS3 2.2.8 Released !](#)

1.2. Origine du projet

Le projet GNS3 (2008) trouve son origine dans le logiciel Open Source [Dynamips](#).

Dynamips (2005) est un **émulateur de routeur Cisco** écrit par Christophe fillot. Il émule les plateformes matérielles C2691, C3620, C3640, C3660, C3725, C3745 et C7206 en faisant fonctionner de véritables images Cisco IOS.

[Dynagen](#) et [Dynagui](#) sont deux autres projets qui s'interfaçent avec Dynamips. GNS3 est en quelque sorte le successeur de ces deux logiciels d'interface qui s'est étendu en supportant d'autres hyperviseurs / émulateurs matériels.

1.3. Considérations sur GNS3 et sur son usage

Un premier avantage est celui du logiciel libre offrant accès à son code, gratuitement, avec une communauté forte et professionnelle. Il se contrôle via une API HTTP REST. Aussi, il s'interface avec un grand nombre d'hyperviseurs qui émulent beaucoup de plateformes. Il permet de capturer du trafic. La charge peut être répartie sur plusieurs serveurs de calcul. Les topologies peuvent être interconnectées au réseau réel.

Si la solution est certainement la plus populaire chez les candidats aux certifications Cisco et autres, il peut être utilisé dans d'autres cas. Tout d'abord, c'est un outil d'apprentissage qui permet de faire tourner des infrastructures représentatives sans aucune contrainte physique. Dans ce cadre, il peut servir pour des PoC (des preuves de concept), pour des démonstrations à des fins de diagnostic ou pour de test d'intégration au réseau réel.

Attention, le soutien d'une charge réelle reste limité par les ressources attribuées aux topologies et à la nature des systèmes utilisés (pour du lab comme des vIOS ou de la production comme un [CSR1000V](#)).

1.4. Alternatives

GNS3 n'est jamais qu'une alternative parmi d'autres, citons-en au moins quatre.

[Packet Tracer](#) est un simulateur (tout a été prévu par des codeurs) qui va très loin. Il nécessite une inscription dans une Académie Cisco. Êtes-vous vous-même un "académicien" régulier ou un formateur d'Académie ? Tant mieux alors. Dans la mesure, Packet Tracer est mieux que rien.

VIRL, veuillez prononcer "viral" est la solution grand public de plateforme de laboratoire du constructeur Cisco Systems (voir plus bas). Son coût est raisonnable par abonnement de plus ou moins 200 EUR par an. Il faut toutefois être riche pour exécuter les topologies : c'est l'usine à gaz Open Source [OpenStack](#) qui les exécute avec un client lourd spécifique. Il est alors conseillé de prendre un serveur distant chez le prestataire Packet. Son intérêt est de pouvoir récupérer légalement des images Cisco IOS.

Du vrai matériel, toujours idéal, mais de moins en moins réaliste.

Enfin, [EVE-NG](#) est certainement le concurrent le plus à la mesure de GNS3. Il dispose d'un modèle commercial différent avec des versions "personnelle" et "professionnelle".

En plus des composants fondamentaux d’une architecture GNS3, d’autres logiciels seront nécessaires tels qu’une librairie et un logiciel de capture (Wireshark), un client VPN (OpenVPN), un logiciel de terminal (Putty), des clients VNC et Spice, ...

2.2. Client GNS3 GUI

GNS3 GUI est le logiciel client qui offre une interface graphique à l’utilisateur final pour gérer ses déploiements. Il s’installe sur l’ordinateur de travail de l’utilisateur. À terme, on imagine se passer de l’installation d’un client “lourd” local au profit d’une interface Web.

2.3. Serveur GNS3 Server

Le serveur prend toute la charge des fonctionnalités du logiciel :

- il s’interface avec les hyperviseurs et émulateurs (principalement Qemu/KVM, Docker et Dynamips) pour exécuter les topologies ;
- il fournit les services de connectivité, de configuration, de déploiement de modèles, de gestion des topologies.

Notons que les instructions de virtualisation “VT-x/AMD-V” du processeur doivent être vues ou activées.

2.4. Types de déploiements du serveur GNS3

S’il est évident que le client doit être installé sur l’ordinateur de travail de l’utilisateur (Windows, Mac OS X, Linux), le serveur peut être exécuté :

- sur le client directement (peu recommandé).
- ou dans une machine virtuelle locale (de moins en moins recommandé) exécutée avec VMware Workstation/-Fusion ou VirtualBox.
- ou dans une machine virtuelle hébergée sur un serveur VMware ESXi local ou distant.
- ou sur un serveur “bare-metal” dédié ou équivalent, ce qui est (**fortement recommandé**).

2.5. Installation de type Bare-Metal

Pour une meilleure expérience, il sera donc recommandé d’utiliser une instance de type Bare-Metal avec un accès natif aux instructions de virtualisation du processeur “VT-x” ou “AMD-V”.

Aussi les performances du serveur GNS3 peuvent se mesurer en fonction du nombre de CPU, de la vitesse des disques (SATA, SSD direct ou non), de l’accès au réseau, de la quantité de mémoire RAM nécessaires pour faire fonctionner une topologie dont tous les éléments sont actifs.

2.6. Dimensionnement du serveur

Un sous-dimensionnement du rapport “machine à démarrer” / “vcpu disponibles” peut ralentir le lancement d’une topologie. De même que les ressources en RAM pourraient être réservées ou s’agréger. Dans cette perspective l’usage des IOSv et des images Docker¹ s’avère être un choix judicieux pour un usage partagé des ressources.

Le dimensionnement nécessaire dépendra des topologies à déployer et des machines utilisées. Pour commencer en CCNA (200-301), on recommandera au minimum :

- 16Go RAM,
- 8 vcpu disponibles
- et 50Go de stockage SSD et plus, le plus rapide possible, en NVMe par exemple.

1. Les projets qui contiennent des images Docker sont difficilement portables.

2.7. Images pour GNS3

Aussi, le fournisseur doit offrir une image Ubuntu 18.04 LTS prête à l'emploi sur le serveur.

Pour des topologies GNS3 avec des périphériques Cisco, on recommandera les images suivantes qui fonctionnent très bien avec qemu/KVM :

- IOSv (L3, 1 vCPU, 512 Mo RAM) pour les routeurs (uniquement L3) (Architecture "x86_64")
- IOSv-L2 (L2, 1 vCPU, 768 Mo RAM) pour les commutateurs (L2/L3) (Architecture "x86_64")
- des conteneurs Docker pour les stations de travail.

Dynamips

Il est toujours possible de lancer un IOS d'une plateforme originale (version IOS 12.4T) qui aurait dû fonctionner sur un routeur physique embarqué. Ce type de solution est très légère si on se contente d'un vieux IOS dans un contexte de ressources limitées.

Qemu/KVM

Pour des exercices qui nécessitent une reproduction plus fidèle de la réalité, on peut monter soi-même ou utiliser l'image d'une machine virtuelle Qemu/KVM. La plupart des "appliances" GNS3 sont des machines Intel 64 bits.

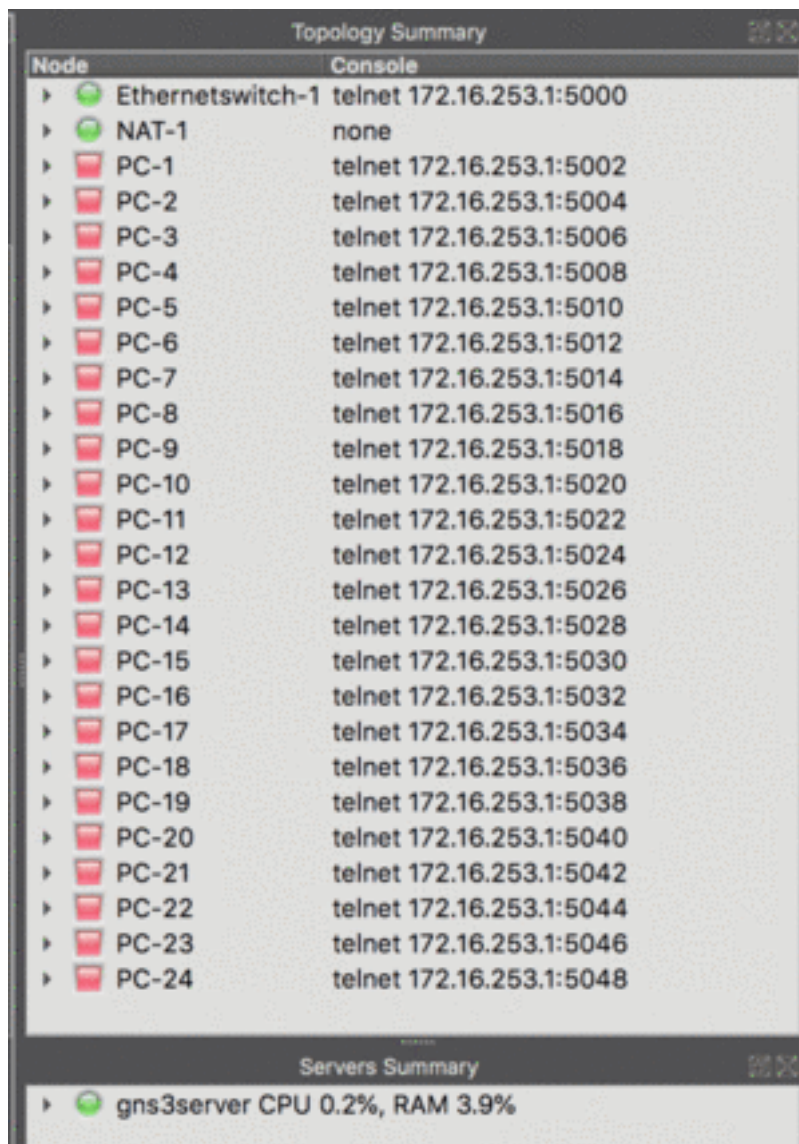
À condition de disposer de ressources suffisantes (plusieurs vCPU et 4G RAM par machine), il est possible de virtualiser Cisco IOS-XE, Cisco IOS-XR ou encore Cisco ASAv.

Conteneurs Docker

Les avantages de la virtualisation par conteneurs et de l'interfaçage de GNS3 avec Docker permettent de profiter pleinement des ressources physiques du serveur avec souplesse et efficacité.

Imaginons le cas d'un grand nombre de stations de travail simples à simuler, des dizaines, et que l'on déploierait par "duplication" à partir d'un conteneur modèle. Chacun de ces éléments sera alors une instanciation logicielle qui partagera toutes les ressources restantes en CPU/RAM et dont la seule charge au démarrage sera un échange DHCP, des échanges Neighbor Discovery et éventuellement du trafic DNS et NTP. La charge disque de chacun de ces conteneurs est presque nulle, mais ce n'est pas un avantage unique de Docker.²

2. Les projets qui contiennent des images Docker sont difficilement portables.



Démarrage de 24 conteneurs

Dans ce cas de figure, les services à offrir doivent avoir la capacité de répondre à cette charge.

Espace disque consommé par les périphériques émulés/simulés

Les machines virtuelles Qemu/KVM sont utilisées en tant que clones liés (selon la terminologie VMware). Les machines virtuelles déployées utilisent un disque “snapshot” qui est le différentiel d’une image de base présente sur le serveur. Cette solution est optimale pour sauvegarder et partager des topologies de lab à usage temporaire.

Par contre, il n’y a pas de différentiel pour les conteneurs déployés. Si l’image de base d’un conteneur est déployée dix fois, il y aura autant d’espace disque consommé.³

2.8. Périphériques intégrés

Parmi les périphériques intégrés, on trouvera :

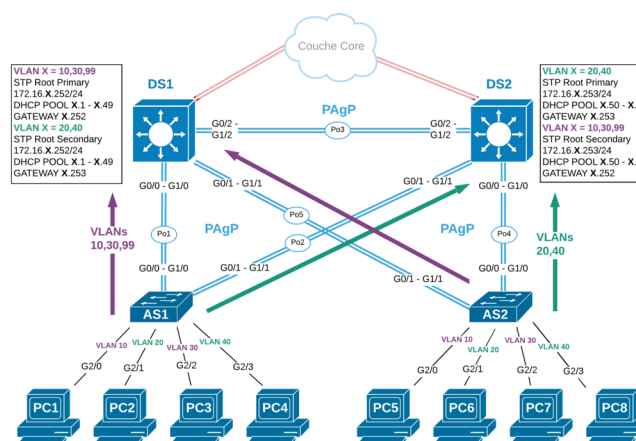
- Hub Ethernet.

3. Les projets qui contiennent des images Docker sont difficilement portables.

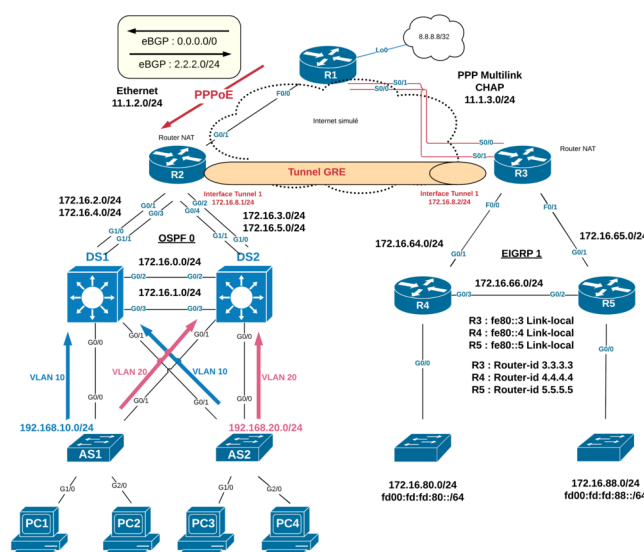
- Commutateur (switch) Ethernet compatible VLAN gérable en ligne de commande.
- Un noeud NAT qui offre la connectivité à partir du réseau 192.168.122.0/24.
- Un noeud VPCS qui offre une connectivité minimale en IPv4 et en IPv6.
- Un commutateur ATM et un commutateur Frame-relay pour les anciennes topologies.

2.9. Exemples de topologies et de leur dimensionnement

Dans les exemples suivants, on fait l'hypothèse d'un démarrage de tous les éléments d'une topologie au même moment.



Nombre	Images	vCPU	RAM
4	IOSv-L2 (switch)	4	3072 Mo
8	Docker (station)	1 minimum	quelque Mo
12	total	4	4 Go



Nombre	Images	vCPU	RAM
4	IOSv-L2 (switch)	4	3072 Mo
3	IOSv (routeur)	3	1536 Mo
2	Dynamips (routeur)	1 minimum	512 Mo
4	Docker (station)	1 minimum	quelque Mo
13	total	8	5,12 Go

2.10. Tableau comparatif des offres des fournisseurs IaaS

Si le prestataire de service Scaleway est illustré ici, il est parfaitement possible de faire la même chose chez d'autres prestataires comme OVH ou Packet.net avec d'autres prix, mais probablement un meilleur service.

Scaleway prépare une nouvelle offre [Bare Metal Dedibox](#). Les instances virtuelles de Scaleway exposent les instructions de virtualisation de telle sorte que GNS3 Server soit fonctionnel sur ces VM, mais il est conseillé alors de prendre un disque NVMe natif comme l'offre [Instances General Purpose \(GP\)](#).

Fournisseur	Offre	Prix	Appréciation
Scaleway	DEV1-L (8Go RAM, 4 vcpu, 80Go NVMe)	15,99 EUR/mois ou 0,032 EUR/heure	+ Choix minimal, mais très bon rapport qualité/prix, instance virtuelle
Scaleway	GP1-XS (16Go RAM, 4 vcpu, 150Go NVMe)	39 EUR/mois ou 0,078 EUR/heure	++ Confortable, bon rapport qualité/prix
Scaleway	GP1-S (32Go RAM, 8 vcpu, 300Go NVMe)	79 EUR/mois ou 0,158 EUR/heure	+++ Très confortable, bon rapport qualité/prix, instance virtuelle
OVH	Serveur Dédié : SP-32-S (32Go RAM, intel Xeon E3-1245 v5 - 4c/8t - 3,5GHz /3,9GHz, 2To SATA)	59,99 EUR/mois	++ Très bon service
Scaleway	GP-BM1-S (64Go RAM, 1× Intel® Xeon E3 1240v6 4C 8T - 3,7 GHz, 2×250 Go SSD)	119,99 EUR/mois ou 0,24 EUR/heure	++ Bare-Metal
Packet.net	c1.small.x86 (32Go RAM, 4 Physical Cores @ 3.5 GHz - 1 × E3-1240 V5, 50Go SSD)	0,40 EUR/heure ou 10 EUR/jour	+ Rolls du Bare-Metal-As-a-Service

3. Installation de GNS3 Server chez Scaleway

Nous partons du principe que vous avez les moyens de vos ambitions en prenant un serveur "Bare-Metal" (entendez "dédié") chez [Scaleway](#) pour exécuter des topologies GNS3.

3.1. Procédure d'installation de GNS3 SERVER

1. Inscription chez Scaleway
2. Créer une instance GP-BM1-S
3. Installation de GNS3 Server
4. Acquisition des images Cisco IOSv
5. Placement des images sur le serveur

3.2. Inscription chez Scaleway

- Inscription chez Scaleway : <https://cloud.scaleway.com/#/signup>
- [Placement d'une clé publique SSH](#)

SSH keys

SSH KEY

ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDIwtTLiaKl+YPV0LjN/DJtS6hYOAEEEXqAS105z8ft6lda+3sG/kd3j90Niepug
G6sanGKRAMvx6520IZDgWbZEC40/7shAoxxbiFXBj3VgPCoKXCNLtTla4nx9hR/Xstzfm6k3/mODiioWV+TwWim9SRO
5ihwF09BhLw4sfTchtUxlLrW6pv0o4ykBPMC90yJl5KrB+7QKwaedbUYwBkXHNDDO47ijZUFS6upF0xhPzWg77den7Ug
z9iMBL/jeOd6WU8eGyEUCUI/OiadSBuY92iyFuksZMCrYtQkX+LgS/RaieMD+lcKMVIwd+vV90Io0a4ndER79kpMrTWg
Rrk1

EDIT

DELETE

SSH Key

NEW SSH KEY

Placement d'une clé publique SSH chez Scaleway

3.3. Créer une instance GP-BM1-S

Prendre une instance **GP-BM1-S** (64Go RAM, 8 cores, 250Go SSD) avec **Ubuntu 18.04 (Bionic)** comme système d'exploitation et la lancer.

3.4. Installation de GNS3 Server

Suivre la procédure décrite dans le document intitulé [Install GNS3 on a remote server](#) (le script d'installation est situé sur [GitHub](#)). Sur le serveur distant GNS3 (gns3-server) :

```
cd /tmp
curl https://raw.githubusercontent.com/GNS3/gns3-server/master/scripts/remote-install.sh \
  > gns3-remote-install.sh
bash gns3-remote-install.sh --with-openvpn
```

A la fin de l'installation, un URL de téléchargement du fichier de connexion OpenVPN est fourni. Cet information fait partie de la bannière qui s'affichera à chaque connexion SSH au serveur.

```
[... sortie ommise ...]
=> Create client configuration
Setup HTTP server for serving client certificate
=> Restart OpenVPN
=> Download http://51.15.90.65:8003/fe718e72-dcb2-11e9-98e6-0007cb0b1f2f/gns3-server.ovpn to setup you\
r OpenVPN client after rebooting the server
```

Selon ce message, on trouvera le fichier de connexion Openvpn au serveur sous l'URL suivant :

`http ://51.15.90.65:8003/fe718e72-dcb2-11e9-98e6-0007cb0b1f2f/gns3-server.ovpn`

Veuillez récupérer ce fichier avec l'extension `.ovpn` sur votre station de travail. Ici avec la commande `wget` :

```
wget http://51.15.90.65:8003/fe718e72-dcb2-11e9-98e6-0007cb0b1f2f/gns3-server.ovpn
```

Enfin, il est recommandé de redémarrer le serveur ("Hard Reset" chez Scaleway).

3.5. Acquisition des images Cisco IOSv

Télécharger les images à partir de son compte Cisco [VIRL](#) (200 EUR/an).

3.6. Placement des images sur le serveur

Placer les images IOSv et Intel dans /opt/gns3/images/QEMU

3.7. Ultimes recommandations

Il y a certainement deux mesures de précaution à prendre si le serveur GNS3 est exposé avec une adresse IP publique : désactiver la mise à disposition du fichier de connexion VPN en HTTP sur le port TCP 8003 et contrer les tentatives harassantes de connexion sur le port SSH TCP 22. On prendra donc garde de désactiver `nginx-light` (ce qui est toujours réversible) qui offre ce service Web. Il est aussi recommandé d'activer le logiciel `fail2ban` pour diminuer la pollution des tentatives de connexions sur le port SSH.

```
apt-get -y install at fail2ban
at tomorrow <<< 'apt-get -y remove nginx-light'
```

3.8. Livre de jeu Ansible pour déployer GNS3 Server

La collection Ansible [Deploy GNS3 Server](#) reprend une série de rôles Ansible qui permettent d'approvisionner des serveurs chez Scalaway ou Packet, de les configurer et d'envoyer les paramètres de connexion à l'utilisateur final par courrier électronique.

4. Comment trouver des images Cisco IOS pour GNS3 ?

Cette étape est facultative si vous n'avez pas besoin d'éprouver des produits Cisco Systems. Les images actuelles fonctionnent en format Intel x86_64, ce qui signifie qu'elles fonctionnent avec un hyperviseur comme qemu/KVM, VMWare ou VirtualBox. Ces images s'obtiennent en payant un abonnement chez Cisco Systems.

4.1. Abonnement Cisco VIRL

Le seul moyen d'obtenir des images IOSv légalement est de prendre un abonnement [VIRL](#).

L'exploitation de la solution native Cisco VIRL nous semble particulièrement [lourde et coûteuse](#). C'est la raison principale de l'exploitation des images Cisco avec GNS3.

4.2. Images Cisco VIRL

Un abonnement VIRL vous permet d'obtenir des [images](#) à exécuter dans GNS3. Il s'agit d'image que l'on peut utiliser légalement pour un usage de test et d'apprentissage à condition d'avoir acheté un abonnement Cisco VIRL.

Soit ces images ne sont pas destinées à supporter du trafic de production comme les images IOSv, soit elles démarreront avec un taux de transfert limité sur les interfaces, comme l'image du CSR 1000v.

On trouvera plus bas un tableau des plateformes avec la version du système d'exploitation et la version de l'abonnement VIRL.

Plateforme	version	VIRL Release
IOSv	15.6(2)T	VIRL 1.3.296 (Aug. 2017 Release)
IOSv	15.6(3)M	VIRL 1.5.145 (March 2018 Release)
IOSv	15.7(3)M (New)	VIRL 1.6.65 (July 2019 Release)
IOSv L2	15.2 (03.2017)	VIRL 1.3.296 (Aug. 2017 Release)
IOSv L2	15.2.1 (06.2018) (New)	VIRL 1.6.65 (July 2019 Release)
IOS XRv	6.1.3	VIRL 1.3.296 (Aug. 2017 Release)
IOS XRv 9000	6.0.1	VIRL 1.3.296 (Aug. 2017 Release)
IOS XRv 9000	6.2.2 image (New)	VIRL 1.5.145 (March 2018 Release)
IOS XRv 9000	6.5.1 image (New)	VIRL 1.6.65 (July 2019 Release)
CSR 1000v	16.5.1b (XE based image)	VIRL 1.3.296 (Aug. 2017 Release)
CSR 1000v	16.6.1 XE-based image (New)	VIRL 1.5.145 (March 2018 Release)
CSR 1000v	16.9.1 XE-based image (New)	VIRL 1.6.65 (July 2019 Release)
NXOSv 7k	7.3.0.1	VIRL 1.3.296 (Aug. 2017 Release)
NXOSv 9k	7.0.3.I6.1	VIRL 1.3.296 (Aug. 2017 Release)
NXOSv 9k	9.2.3 (Nexus 9000)	VIRL 1.6.65 (July 2019 Release)
ASAv	9.7.1	VIRL 1.3.296 (Aug. 2017 Release)
ASAv	9.8.2 (New)	VIRL 1.5.145 (March 2018 Release)
ASAv	9.9.2 (New)	VIRL 1.6.65 (July 2019 Release)

4.3. Images Cisco pour les Labs

Les images utilisées Qemu/KVM dans ce support sont les suivantes.

Périphériques	Images Qemu/KVM	Commentaire
Routeur Cisco IOSv	<code>vios-adventerprisek9-m.vmdk.SPA.156-2.T</code> ou <code>vios-adventerprisek9-m.vmdk.SPA.157-3.M3</code> avec <code>IOSv_startup_config.img</code>	VIRL
Commutateur Cisco IOSv L2/L3	<code>vios_l2-adventerprisek9-m.03.2017.qcow2</code> ou <code>vios_l2-adventerprisek9-m.SSA.high_iron_-20180619.qcow2</code>	VIRL
Poste de travail L2 à L7	<code>container Docker gns3/ubuntu :bionic</code>	Ubuntu GNS3 Docker Hub
Pare-feu ASAv	<code>asav971.qcow2</code>	VIRL

4.4. Alternative avec des images Dynamips

Sans abonnement VIRL, il est difficile de prendre possession de ces images. Toutefois, il est possible d'obtenir un routeur IOS en version 12.4 plus facilement, car celui-ci n'est plus maintenu depuis longtemps. On fait alors fonctionner ces machines "virtuelles" avec l'émulateur Dynamips. Les images Dynamips suivantes sont fonctionnelles :

Périphériques	Images dynamips
Routeur IOS 12.4 (petit)	<code>c3725-adventerprisek9-mz.124-15.T14</code>
Routeur IOS 12.4 + CME (VoIP)	<code>c3745-adventerprisek9_ivs-mz.124-15.t8</code>
Routeur IOS 12.4 (Mainstream)	<code>c7200-adventerprisek9-mz.124-24.T5</code>

5. Installation de GNS3 GUI et connexion au serveur

L'étape suivante est d'installer la solution "client" sur le poste de travail de l'utilisateur. En résumé, le logiciel client GNS3-GUI va se connecter au serveur GNS3-Server à travers un tunnel VPN OpenVPN. Pour que les deux éléments puissent communiquer ensemble, le tunnel VPN doit être monté.

5.1. Procédure d'installation de GNS3-GUI

La procédure consiste à suivre quatre étapes :

1. Télécharger et installer le logiciel OpenVPN.

2. Établir la connexion OpenVPN avec votre serveur GNS3
3. Installer la version GUI de GNS3
4. Importer des “appliances” (les modèles de machines virtuelles)

5.2. Téléchargement et installation d'OpenVPN

- Pour Windows : <https://openvpn.net/index.php/open-source/downloads.html>
- Pour Mac OS X : <https://tunnelblick.net/>
- Pour Linux : `apt -y install openvpn || yum -y install openvpn`

Il est probablement plus que nécessaire de redémarrer les stations Windows et Mac, car de nouveaux pilotes de périphériques ont été installés.

5.3. Connexion OpenVPN

Dans l'étape d'installation du serveur GNS3-Server, un fichier de connexion VPN a été produit et récupéré : `http://$MY_IP_ADDR:8003/$UUID/$HOSTNAME.ovpn` où la variable `$MY_IP_ADDR` est l'adresse IP du serveur.

Il est alors nécessaire de lancer le client Openvpn graphique ou en ligne de commande avec ce fichier de configuration pour établir la connexion.

Vous pouvez vérifier la connectivité vers votre serveur adressé en `172.16.253.1` à l'autre bout du tunnel :

```
ping -c1 172.16.253.1
```

Avec le résultat attendu :

```
PING 172.16.253.1 (172.16.253.1): 56 data bytes
64 bytes from 172.16.253.1: icmp_seq=0 ttl=64 time=41.023 ms

--- 172.16.253.1 ping statistics ---
1 packets transmitted, 1 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 41.023/41.023/41.023/0.000 ms
```

5.4. Téléchargement et installation de GNS3-GUI

Le logiciel se télécharge à partir de cette adresse : <https://github.com/GNS3/gns3-gui/releases/> à partir de laquelle on trouve des différentes archives pour Windows, Mac, Linux et les fabrications des machines virtuelles GNS3-VM.

Pour Windows et Mac OS X

Sur des systèmes d'exploitation comme Windows et Mac OS X, l'installation est très intuitive, car elle consiste uniquement à choisir “Next”, excepté qu'il n'est pas nécessaire d'installer les produits commerciaux quand bien même ils sont très intéressants (SolarWinds Response Time Viewer).

On trouvera une aide dans le document intitulé [Windows Installation](#).

Pour Linux (Debian)

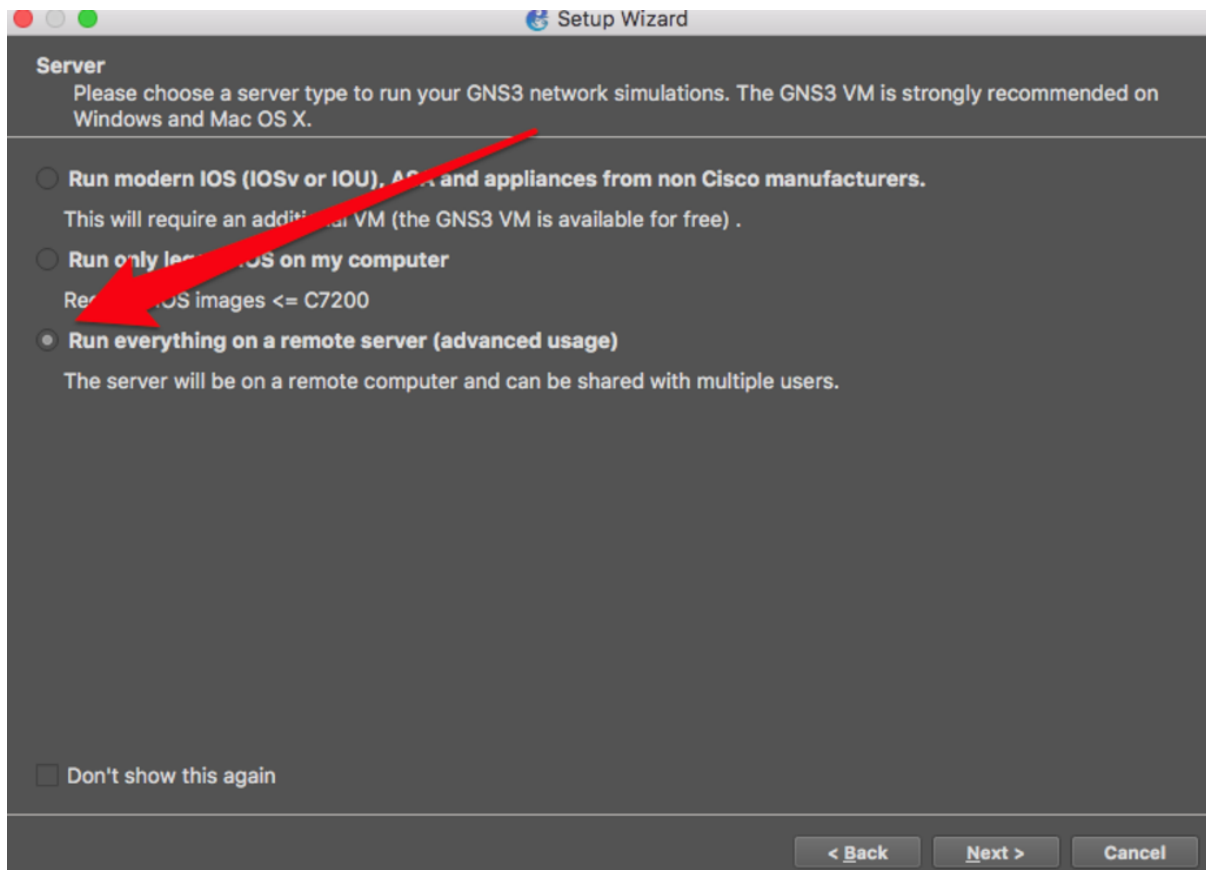
Le document suivant indique la procédure d'installation de GNS3-gui sous Linux : [GNS3 Installation on Linux](#).

On trouvera ici une procédure d'installation pour Centos 7 : [Gist install-gns3-centos7.sh](#)

Pour une installation de GNS3-GUI sous Linux, il est conseillé de vérifier la présence de Wireshark.

Choix de la configuration

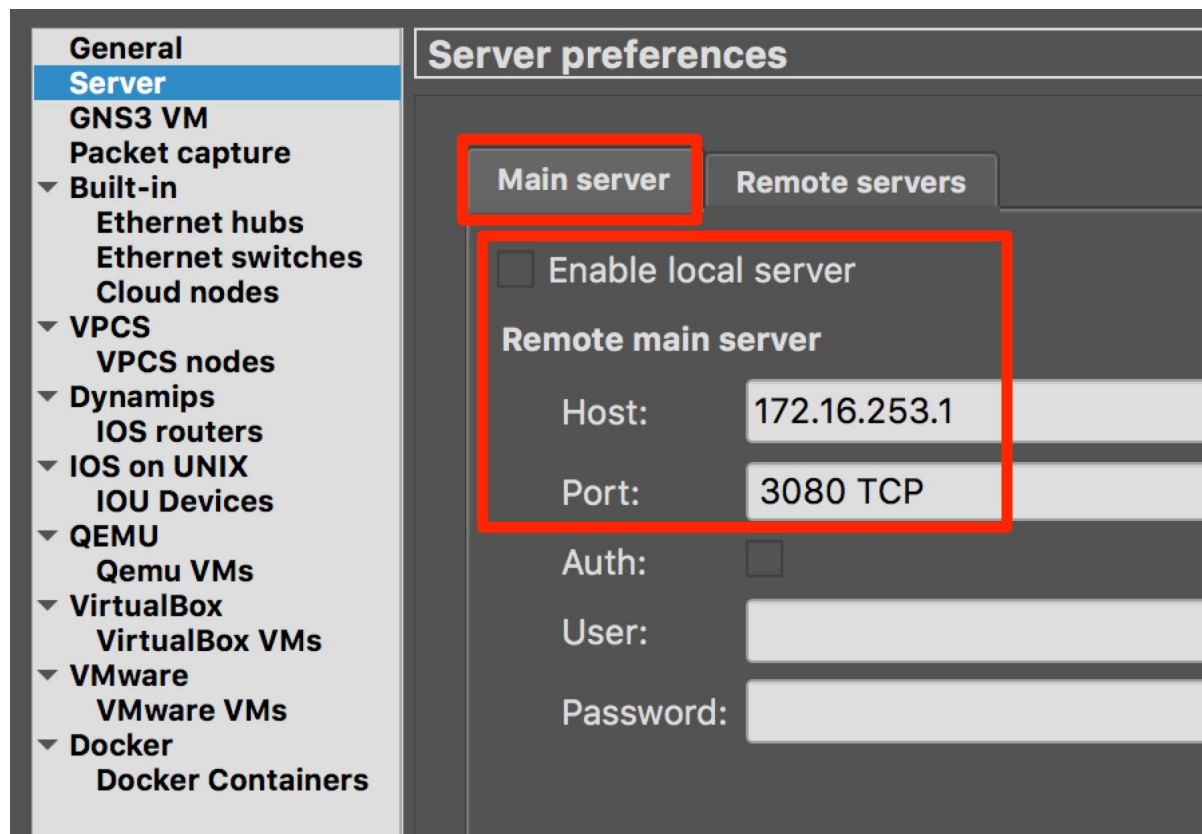
Lors de l'installation choisir une configuration "Run everything on a remote server (advanced usage)".



Fenêtre Run everything on a remote server (advanced usage)

Ce serveur distant est à l'autre bout du tunnel VPN. Les paramètres à indiquer sont :

- Adresse : 172.16.253.1
- Port : 3080



Connexion au serveur à travers le tunnel VPN

5.5. Connexion à l'interface Web

A partir de la version 2.2, on peut tester l'interface de gestion à travers un navigateur Web (Firefox ou Chrome) et même accéder aux consoles dans l'interface Web. Voyez vous-même : <http://172.16.253.1:3080/static/web-ui/bundled>.

6. Appliances GNS3

6.1. Définition d'une appliance GNS3

Un template (un modèle) d'*appliance* est un fichier de définition d'une machine virtuelle qui référence les paramètres d'interfaces, la RAM, les CPUs, la console et des images disque.

Depuis la version 2.1, les modèles des "appliances" sont directement disponibles à travers l'interface GNS3 GUI ce qui facilite le déploiement en classe de formation. Il est conseillé d'utiliser l'*appliance* Ubuntu Docker comme station de travail en attendant la disponibilité d'un Windows Core en image Docker par exemple.

Leur emplacement sur le serveur est situé ici :

```
/usr/share/gns3/gns3-server/lib/python3.6/site-packages/gns3server/appliances/
```

Pour utiliser une "appliance", il est nécessaire que l'image soit placée sur le serveur.

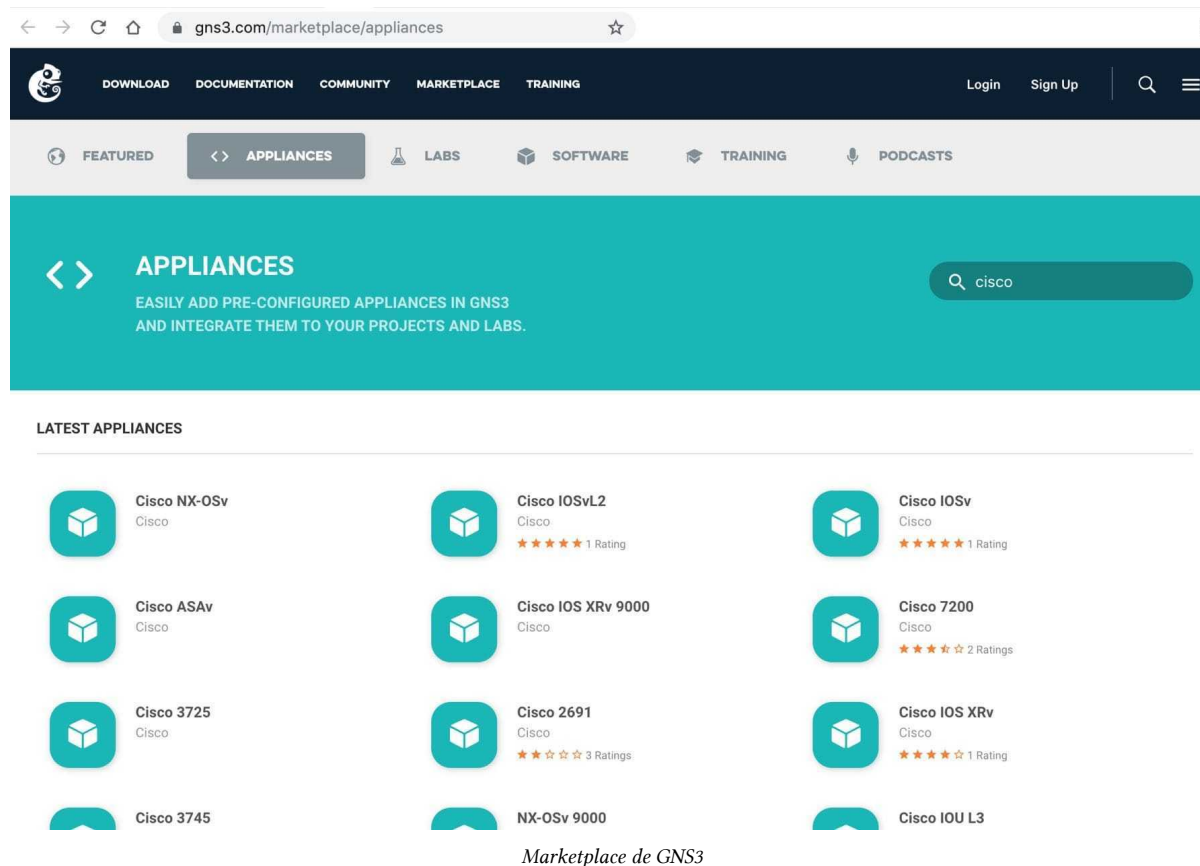
Ces images pourraient être placées d'avance sur le serveur GNS3, ce qui est recommandé. Dans le cas contraire, le cas échéant le fichier d'appliance indique un URL de téléchargement avec la possibilité de la placer à travers le canal de contrôle HTTP éventuellement encapsulé dans le tunnel OpenVPN (bonne chance si l'image dépasse quelques dizaines de Mo).

[à compléter]

Ensuite, l'“appliance” devrait être “installée” ...

6.2. Marketplace GNS3

Le [Marketplace de GNS3](#) donne une très bonne idée de la diversité des plateformes que l'on peut émuler avec le logiciel.



On remarquera :

- [OpenWrt](#) et [OpenWrt Realview](#)
- [pfSense](#), mais aussi [OPNsense](#) qui supporte EIGRP, OSPF, BGP et IPv6.
- [Open vSwitch](#)
- Juniper, Microsoft, Cisco, F5, Mikrotik, Arista, HPE, Alcatel, Smoothwall, Brocade, Fortinet, ntopng, ...
- Diverses appliances basées Docker

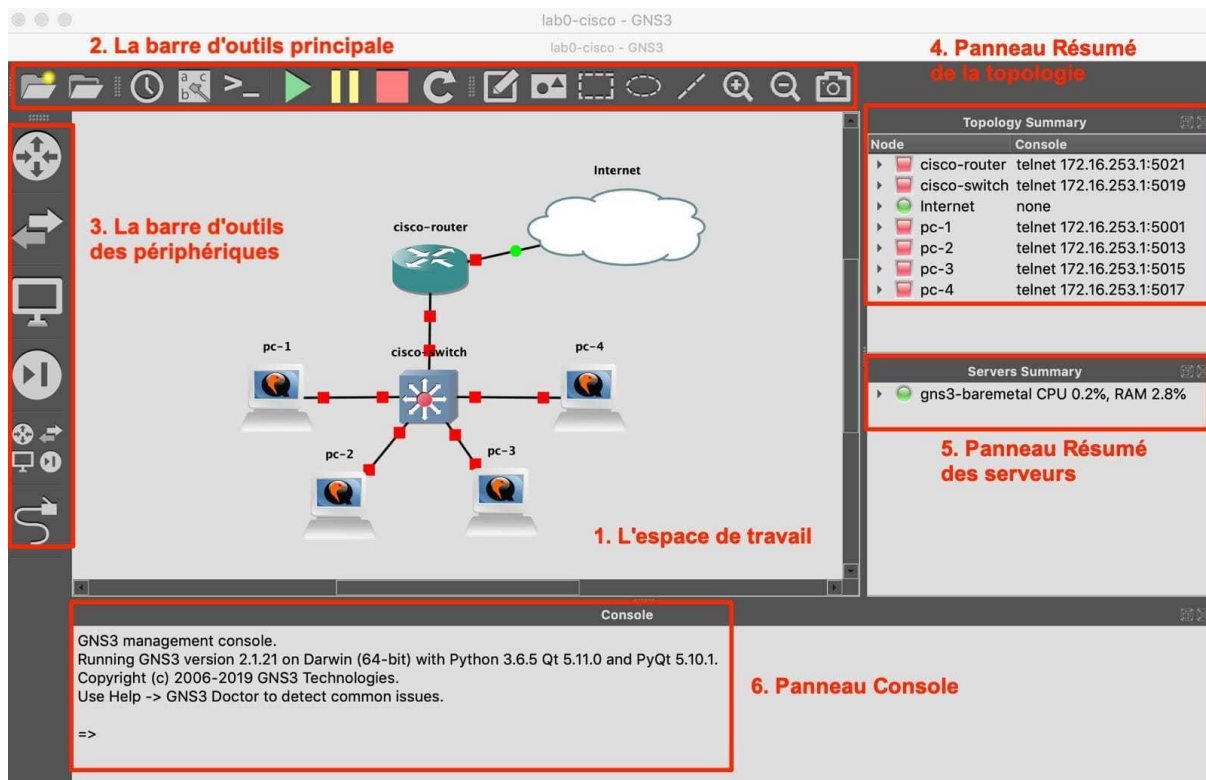
7. Création et manipulation d'un projet GNS3

L'usage de GNS3 est très intuitif : On déplace des périphériques sur l'espace de travail, on tire les câbles de connexion, on allume les périphériques de la topologie et puis on peut accéder aux consoles de gestion.

Page d'aide : [Your First GNS3 Topology](#)

L'interface graphique est composée de six éléments :

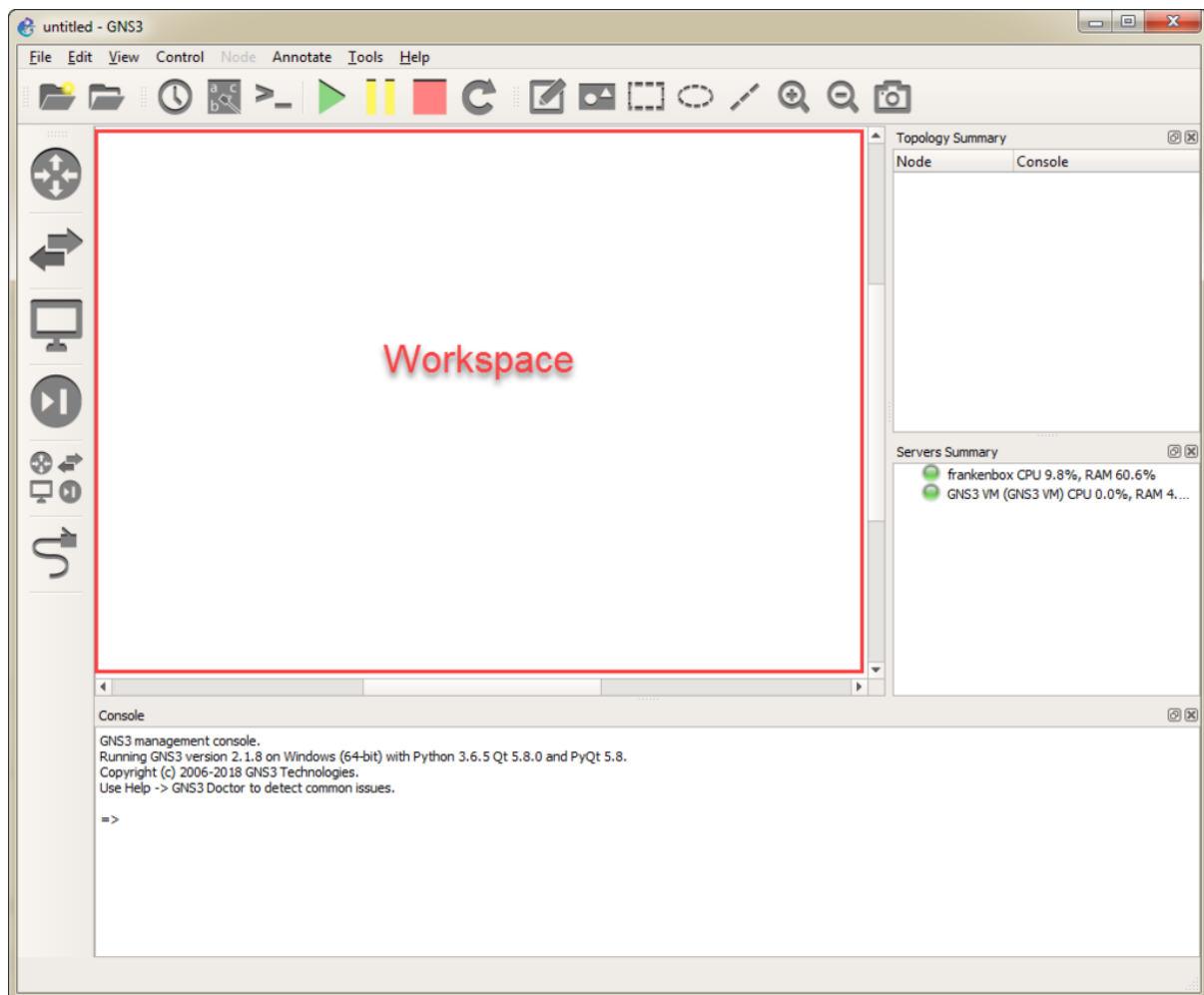
1. Espace de travail (Workspace)
2. Barre d'outils principale (Toolbar)
3. Barre de périphériques (Devices Toolbar)
4. Panneau Résumé de la topologie
5. Panneau Résumé des serveurs
6. Panneau Console du contrôleur



Interface graphique de GNS3

7.1. Espace de travail (Workspace)

L'espace de travail représente la topologie et permet de contrôler à la souris les différents périphériques à partir d'un clic droit.

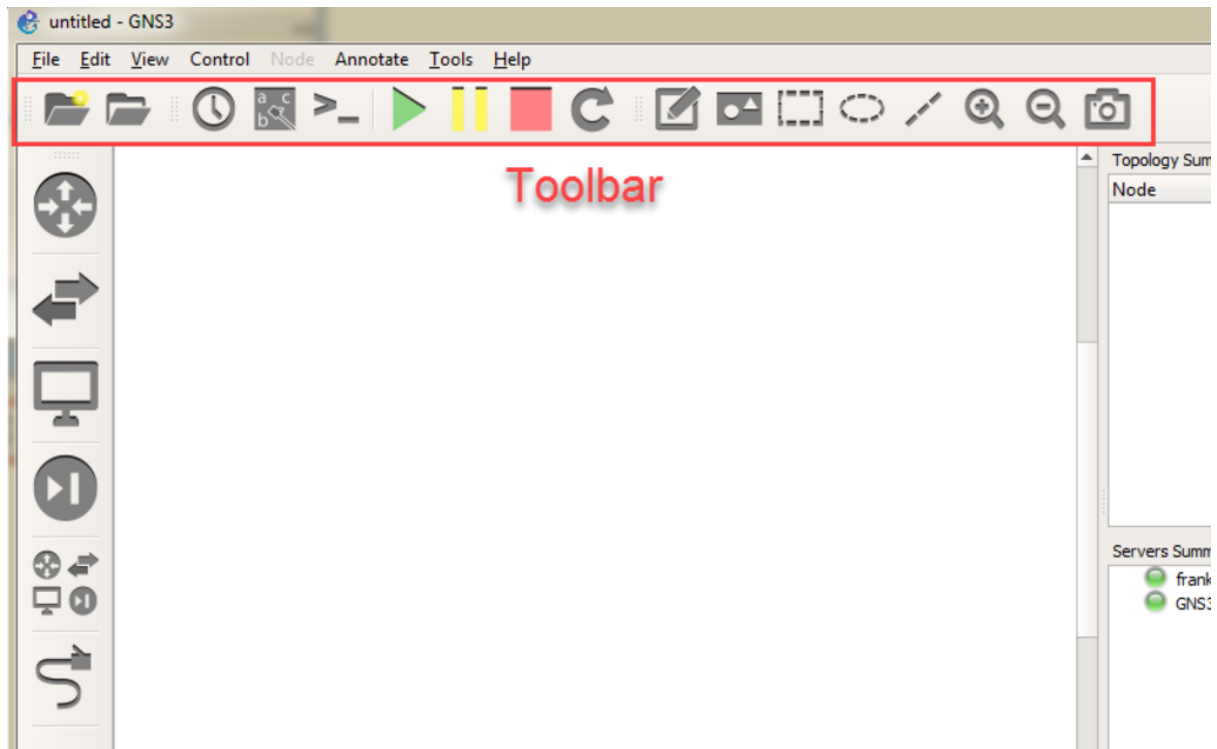


Espace de travail (Workspace) GNS3

7.2. Barre d'outils (Toolbar)

Cette barre d'outils générale permet de :

- créer une nouvelle topologie
- ouvrir une nouvelle topologie
- gérer les instantanés
- Ouvrir la console, démarrer, mettre en pause, arrêter tous les périphériques de la topologie
- Ajouter du texte, des images, des figures à la topologie
- Prendre une image graphique de la topologie

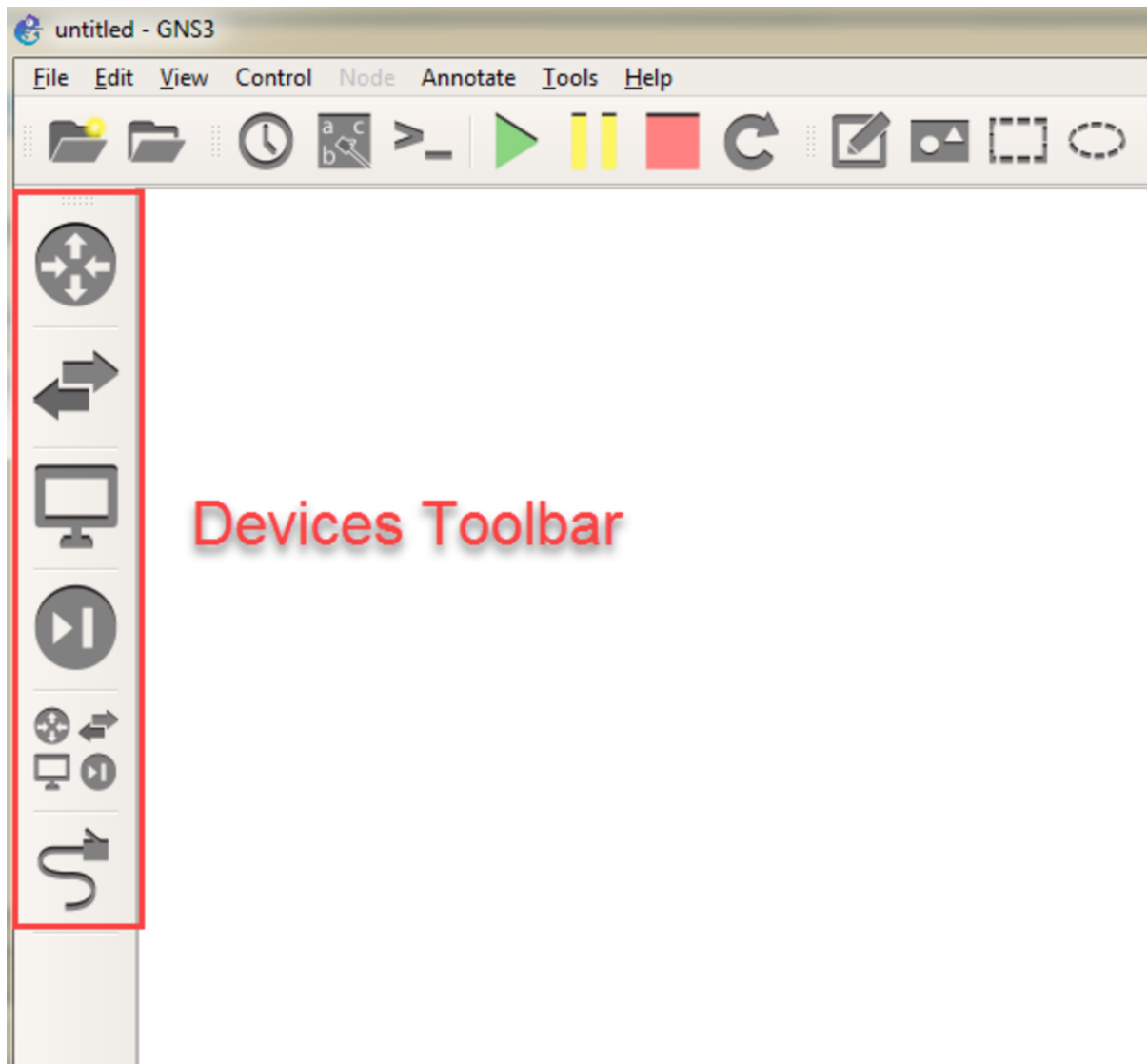


Barre d'outils (Toolbar) GNS3

7.3. Barre de périphériques (Devices Toolbar)

La barre de périphérique permet de gérer les appliances et de déployer les périphériques et les connexions :

- Les routeurs
- Les commutateurs
- Les hôtes terminaux
- Les pare-feux
- Tous les périphériques
- Les connexions

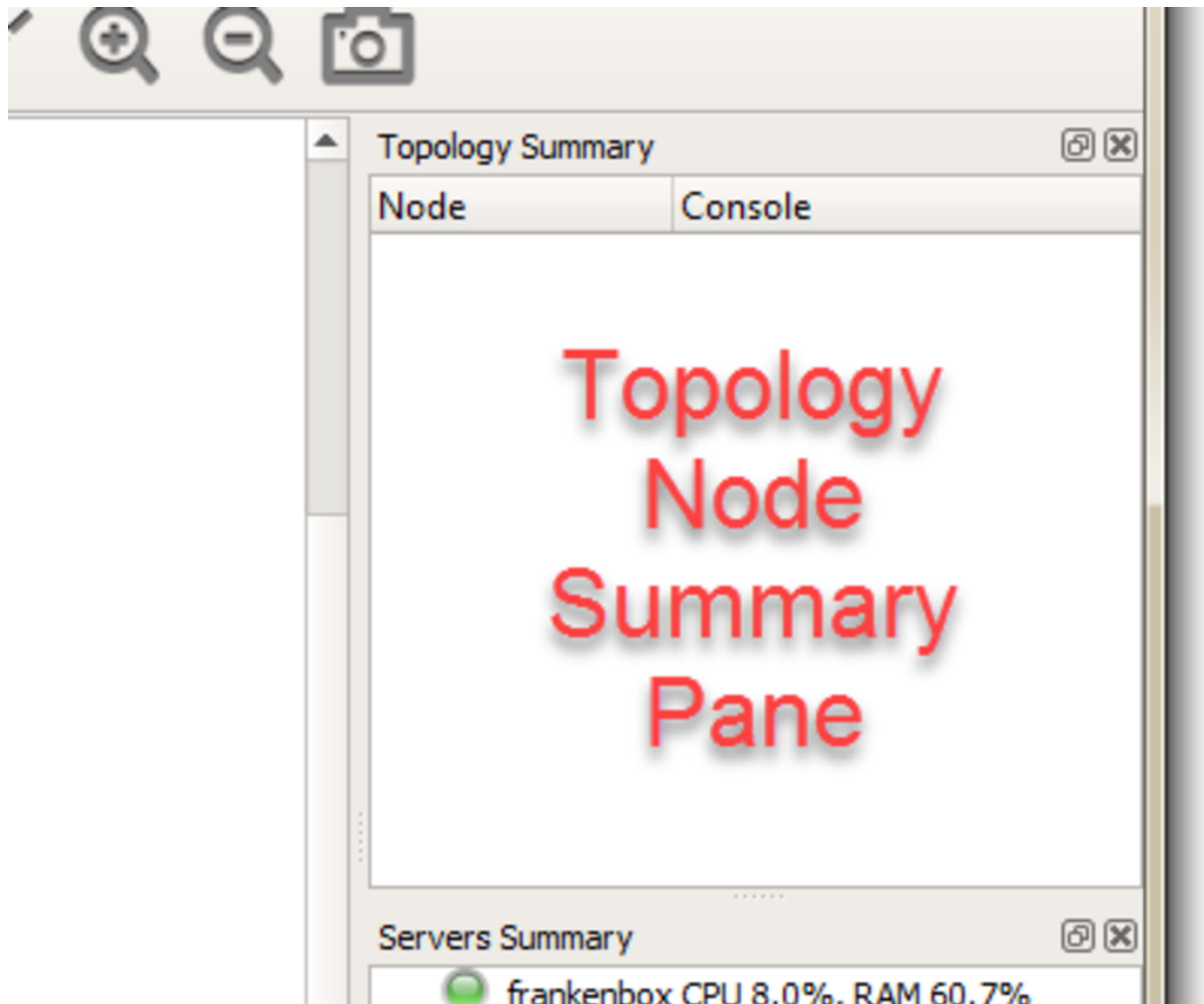


Barre de périphériques (Devices Toolbar) GNS3

7.4. Panneau Résumé de la topologie (Topology Node Summary Pane)

Ce panneau donne la liste des périphériques avec l'adresse et le port d'accès de la console texte (via telnet). Un clic droit sur un périphérique offre les options de contrôle.

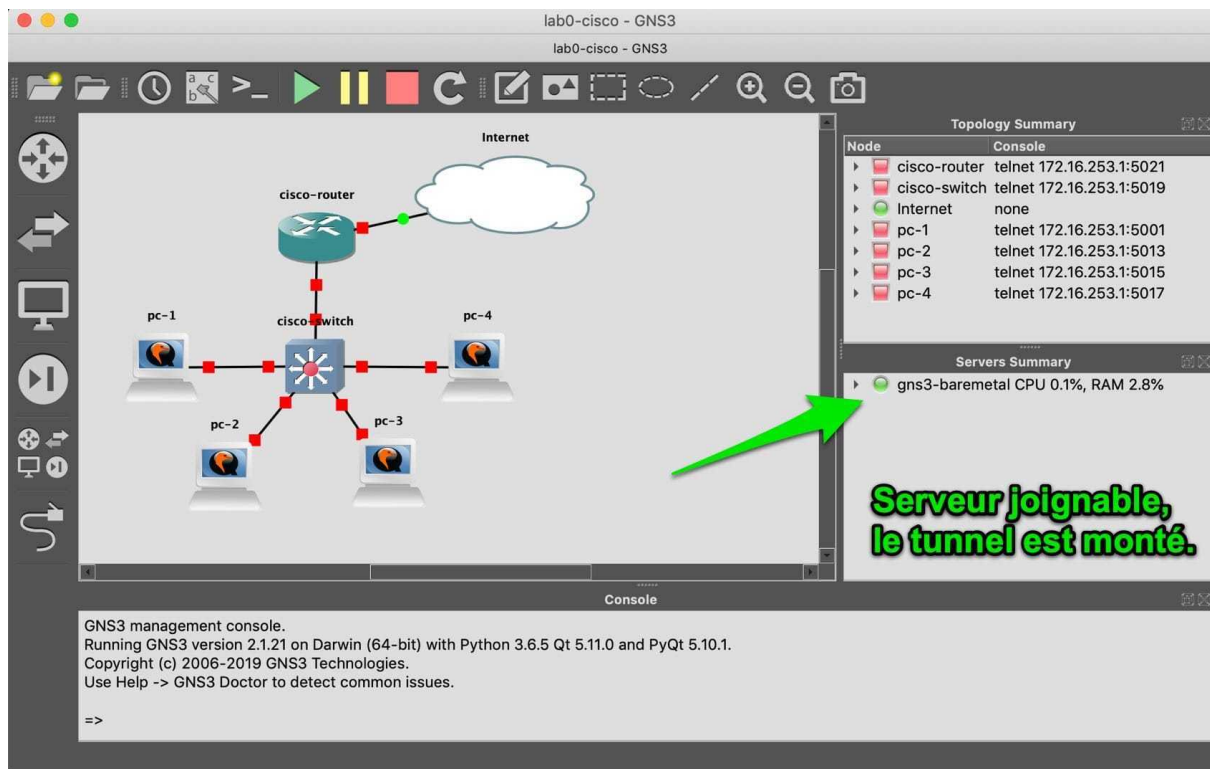
On peut aussi visualiser les connexions entre les périphériques et lancer des captures à visualiser dans wireshark.



Panneau Résumé de la topologie (Topology Node Summary Pane) GNS3

7.5. Panneau Résumé des serveurs (Servers Summary Pane)

Le panneau Résumé des serveurs donne la liste des serveurs référencés dans la configuration GNS3-gui, leur capacité en CPU et en RAM.

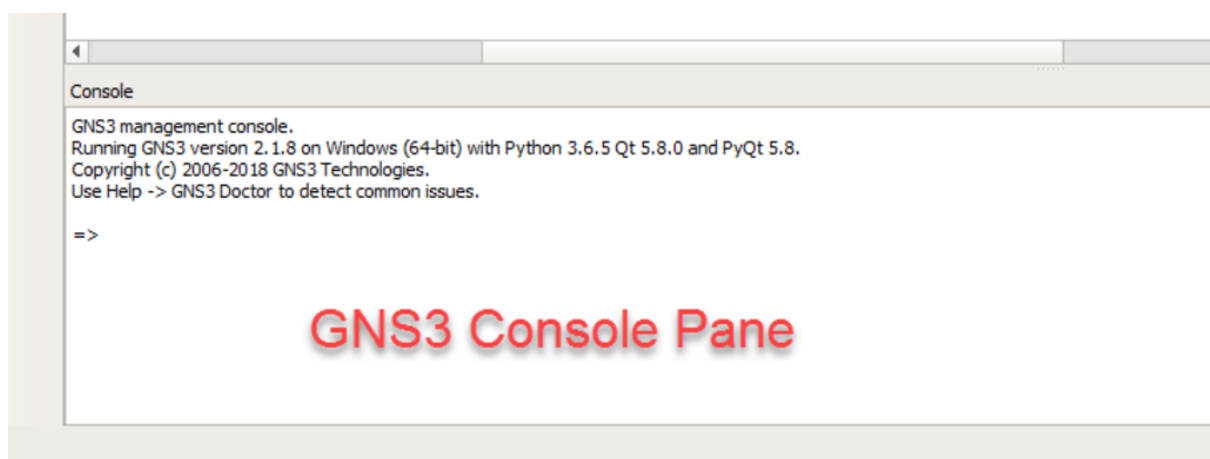


Panneau Résumé des serveurs (Servers Summary Pane) GNS3

En voyant vos serveurs disponibles, vous pouvez déduire que la connectivité à travers le tunnel OpenVPN est établie et que le serveur est prêt à recevoir les commandes venant du client.

7.6. Panneau Console du contrôleur (Console Pane)

Il s'agit de la console texte qui permet de contrôler en ligne de commande la topologie.



Panneau Console du contrôleur (Console Pane) GNS3

GNS3 management console.

Running GNS3 version 2.1.21 on Darwin (64-bit) with Python 3.6.5 Qt 5.11.0 and PyQt 5.10.1.

Copyright (c) 2006-2019 GNS3 Technologies.

Use Help -> GNS3 Doctor to detect common issues.

=> help

Documented commands (type help <topic>):

=====

console debug help log reload show start stop suspend version

=> show ?

Show detail information about every device in current lab:

show device

Show detail information about a device:

show device <device_name>

=> show device

Nat device Internet is always-on

This is a node for external connections

Device run on gns3-baremetal

Port nat0 connected to cisco-router on port Gi0/1

QEMU VM pc-1 is stopped

Node ID is 2, server's node ID is 3b6840cf-d0f9-48dc-80b5-d0464baeea37

QEMU VM's server runs on gns3-baremetal

Console is on port 5001 and type is telnet

Ethernet0 connected to cisco-switch on port Gi0/1

QEMU VM pc-2 is stopped

Node ID is 3, server's node ID is 7bc22b33-4536-4050-8c6e-13c24a2cb979

QEMU VM's server runs on gns3-baremetal

Console is on port 5013 and type is telnet

Ethernet0 connected to cisco-switch on port Gi0/2

QEMU VM pc-3 is stopped

Node ID is 4, server's node ID is e40f3a4e-cfe7-44b8-9472-f3be26c3a6e3

QEMU VM's server runs on gns3-baremetal

Console is on port 5015 and type is telnet

Ethernet0 connected to cisco-switch on port Gi0/3

QEMU VM pc-4 is stopped

Node ID is 5, server's node ID is 6f803b43-dac0-4d9b-8807-d17504b0b749

QEMU VM's server runs on gns3-baremetal

Console is on port 5017 and type is telnet

Ethernet0 connected to cisco-switch on port Gi1/0

QEMU VM cisco-switch is stopped

Node ID is 6, server's node ID is 32889769-8376-491a-b0df-e48f470304a9

QEMU VM's server runs on gns3-baremetal

Console is on port 5019 and type is telnet

Usage: There is no default password and enable password. There is no default configuration present.

Gi0/0 connected to cisco-router on port Gi0/0

Gi0/1 connected to pc-1 on port Ethernet0

Gi0/2 connected to pc-2 on port Ethernet0

```
Gi0/3 connected to pc-3 on port Ethernet0
Gi1/0 connected to pc-4 on port Ethernet0
Gi1/1 is empty
Gi1/2 is empty
Gi1/3 is empty
Gi2/0 is empty
Gi2/1 is empty
Gi2/2 is empty
Gi2/3 is empty
Gi3/0 is empty
Gi3/1 is empty
Gi3/2 is empty
Gi3/3 is empty

QEMU VM cisco-router is stopped
Node ID is 7, server's node ID is a8fc8d8d-d50a-42eb-a416-75b692aa9ef1
QEMU VM's server runs on gns3-baremetal
Console is on port 5021 and type is telnet
Usage: There is no default password and enable password. There is no default configuration present.
Gi0/0 connected to cisco-switch on port Gi0/0
Gi0/1 connected to Internet on port nat0
Gi0/2 is empty
Gi0/3 is empty

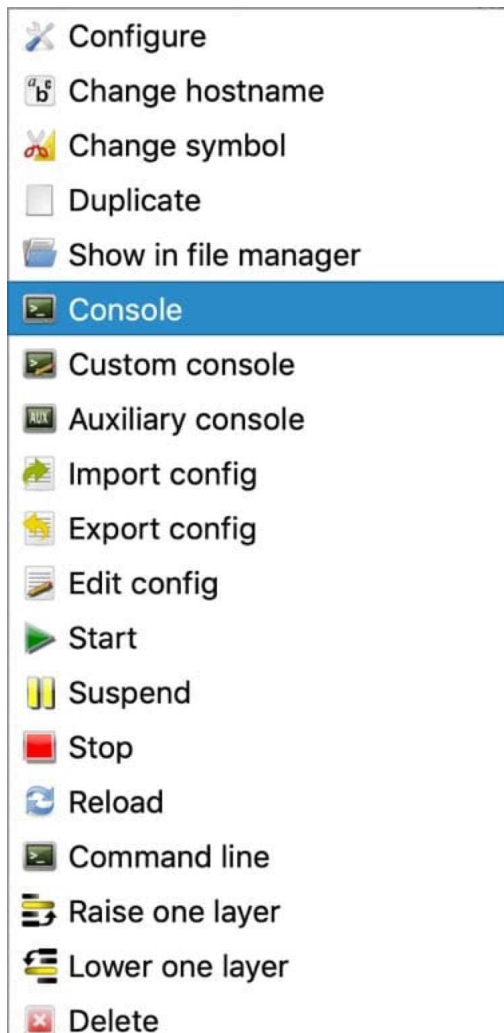
=>
```

7.9. Connexion à la console

Pour se connecter à la console d'un périphérique, il suffit de double-cliquer sur le périphérique ou de faire clic droit et choisir "Console". Dans l'interface Web, un clic droit sur le périphérique permet de choisir une console locale ou dans le navigateur.

7.10. Option de gestion

Un clic droit sur un périphérique de type "Docker" par exemple offre ceci :



Clic droit sur un périphérique GNS3

On trouvera différentes options de gestion intéressantes :

- Configuration et fichier de configuration
- Changement de nom
- Changement de diagramme
- Duplication
- Console
- Démarrage, suspension, arrêt, suppression
- ...

8. Sujets avancés

8.1. Configuration de GNS3

Edit/Preferences :

[à compléter]

8.2. Configuration des consoles

- Texte (Telnet)
- Graphique (VNC)
- Graphique (Spice)

8.3. Capture du trafic et analyse Wireshark

[à compléter]

8.4. Configurer les paramètres réseau d'un container Docker

[à compléter]

8.5. Construction d'une "appliance" et d'une image personnalisée

[à compléter]

- Méthodes de création
- Disques *qcow2* ou *vmdk*
- fichier d'appliance *gns3a* (xml)
- Exemples Windows

8.6. Import / Export d'un projet "portable"

[à compléter]

8.7. Connexion du PC local à la topologie distante

Il existe un périphérique qui s'appelle "NAT node" et qui offre un service de connectivité vers l'Internet, avec DHCP et DNS. Il s'agit d'un routeur NAT virtuel natif par défaut à la solution Libvirt sous Linux. Normalement ce "nuage" est inaccessible de l'Internet, mais il est possible de le joindre assez facilement à travers le tunnel VPN.

Pour joindre le nuage "NAT" adressé en 192.168.122.0/24 dans une topologie, il faut que le client connaisse la route IPv4 à travers le tunnel OpenVPN. Soit le client crée sa route lui-même, soit la configuration est poussée à partir du serveur. Faut-il aussi que le routage entre le tunnel et ce commutateur virtuel soit activé sur le serveur GNS3.

Sur l'ordinateur client, on crée une route statique pour joindre le réseau NAT à travers le tunnel :

```
sudo ip route add 192.168.122.0/24 dev tun0
```

D'une autre manière, on peut indiquer les routes à pousser auprès du client dans le fichier de configuration du serveur Openvpn (/etc/openvpn/udp1194.conf) :

```
echo 'push "route 192.168.122.0 255.255.255.0"' >> /etc/openvpn/udp1194.conf
```

Pour activer le routage entre le tunnel Openvpn et le nuage NAT sur le serveur GNS3-Server, il est nécessaire d'exécuter quelques commandes iptables :


```
iptables -D FORWARD -o virbr0 -j REJECT --reject-with icmp-port-unreachable
iptables -D FORWARD -i virbr0 -j REJECT --reject-with icmp-port-unreachable
iptables -A FORWARD -d 192.168.122.0/24 -s 172.16.253.0/24 -j ACCEPT
iptables -A FORWARD -s 192.168.122.0/24 -d 172.16.253.0/24 -j ACCEPT
```

On peut alors à partir de l'ordinateur GNS3 client joindre le routeur/NAT du nuage et tout périphérique voisin dans le réseau 192.168.122.0/24, comme cela par exemple :

```
ping -c1 192.168.122.1
PING 192.168.122.1 (192.168.122.1) 56(84) bytes of data.
64 bytes from 192.168.122.1: icmp_seq=1 ttl=64 time=1.12 ms

--- 192.168.122.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.122/1.122/1.122/0.000 ms
```

8.8. API HTTP REST de GNS3 Server

À partir de l'ordinateur client connecté au tunnel VPN, vous pouvez manipuler l'API HTTP REST du serveur GNS3, par exemple ici pour obtenir la version courante du serveur :

```
uri="http://172.16.253.1:3080/v2"
```

```
curl $uri/version
```

```
{
  "local": false,
  "version": "2.2.8"
}
```

De la même manière `curl $uri/projects` offrira en format JSON la liste des projets.

Voyez la documentation de l'API : [GNS3 API Documentation : Sample sessions using curl](#).

Voici un exemple de script Bash⁴ qui permet de dupliquer un projet dont vous connaissez l'identifiant sur le serveur :

```
#!/bin/bash

project="$1"
name="$2"

curl --location --request POST "http://172.16.253.1:3080/v2/projects/${project}/duplicate" \
--header 'Content-Type: application/json' \
--data-raw '{"name": "'"$name"'"}'
```

Dans cet exemple, on propose la duplication du projet `e0147533-1edf-4575-bc75-77fe0f281ee5` sous le nouveau nom `project1` :

```
bash gns3-duplicate-project.sh e0147533-1edf-4575-bc75-77fe0f281ee5 project1
```

Cette dernière commande offre cette sortie :

```
4. gns3-duplicate-project.sh
```

```
{
  "auto_close": true,
  "auto_open": false,
  "auto_start": false,
  "drawing_grid_size": 25,
  "filename": "project1.gns3",
  "grid_size": 75,
  "name": "project1",
  "path": "/home/gns3/projects/c7740fba-576b-4a8e-b82e-ce3b0b9b7550",
  "project_id": "c7740fba-576b-4a8e-b82e-ce3b0b9b7550",
  "scene_height": 1000,
  "scene_width": 2000,
  "show_grid": false,
  "show_interface_labels": true,
  "show_layers": false,
  "snap_to_grid": false,
  "status": "closed",
  "supplier": null,
  "variables": null,
  "zoom": 120
}
```

8.9. GNS3 as a Service

Le seul fournisseur de service de serveurs GNS3 à la demande connu sur le marché est [CloudMyLab](#). Le service est “tout compris”, notamment la fourniture d’images Cisco Systems.

Le coût peut toutefois être un frein. Voyez vous-même.

Pricing				
What is GNS3				
What's included?				
Why and who?				
GNS3-TRIAL				
GNS3-How to Videos				
	GNS3-Tiny	GNS3-Small	GNS3-Medium	GNS3-Large
	\$30/Month	\$50/Month	\$75/Month	\$135/Month
	4 vCPU	6 vCPU	8 vCPU	12 vCPU
	8GB RAM	16 GB RAM	24 GB RAM	48 GB RAM
	20 GB SSD	40 GB SSD	60GB SSD	80 GB SSD
	1 Management PC	1 Management PC	1 Management PC	1 Management PC
	Unlimited Bandwidth	Unlimited Bandwidth	Unlimited Bandwidth	Unlimited Bandwidth
	1 Public IP	1 Public IP	1 Public IP	1 Public IP
	24/7 Rack Support	24/7 Rack Support	24/7 Rack Support	24/7 Rack Support
	Add to cart	Add to cart	Add to cart	Add to cart
	GNS3-XLARGE	GNS3-XXL	GNS3-XXXL	GNS3-4XL
	\$160/Month	\$225/Month	\$325/Month	\$450/Month
	14 vCPU	16 vCPU	20 vCPU	32 vCPU
	64 GB RAM	80 GB RAM	96 GB RAM	128 GB RAM
	100 GB SSD	100 GB SSD	160GB SSD	160 GB SSD
	1 Management PC	1 Management PC	1 Management PC	1 Management PC
	Unlimited Bandwidth	Unlimited Bandwidth	Unlimited Bandwidth	Unlimited Bandwidth
	1 Public IP	1 Public IP	1 Public IP	1 Public IP
	24/7 Rack Support	24/7 Rack Support	24/7 Rack Support	24/7 Rack Support
	Add to cart	Add to cart	Add to cart	Add to cart

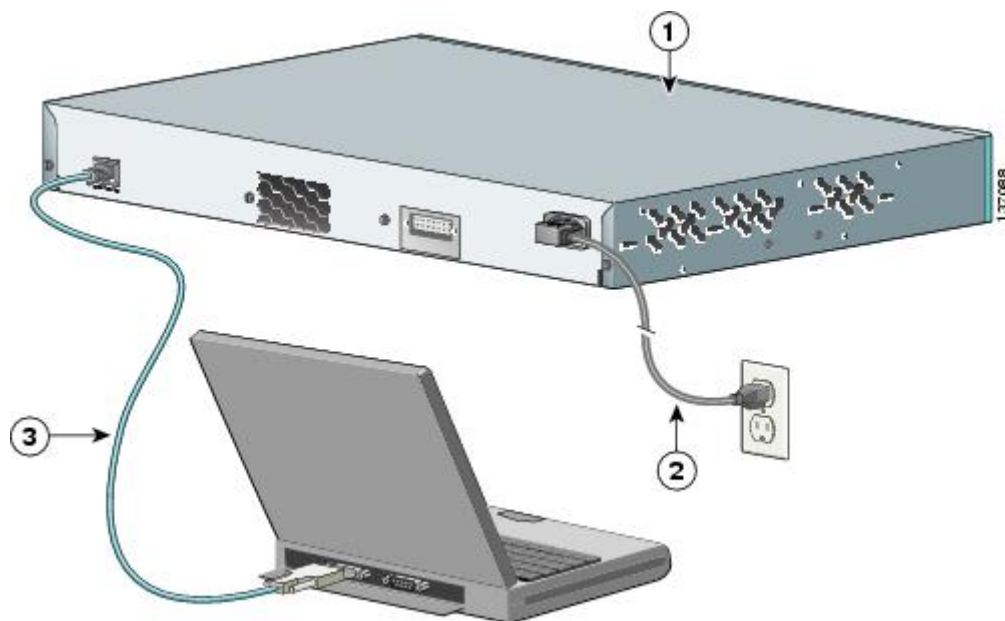
Offres GNS3 Hosted Service CloudMyLab

8. Connexion à un commutateur Cisco

Ce chapitre explique comment se connecter à un commutateur Cisco en console physique et comment reprendre la main sur un commutateur dont a perdu le mot de passe via un *password recovery*.

1. Connexion série / console

- Câble inversé (roll-over) du port COM1 du PC au routeur sur le port console.
- Lancer un logiciel d'émulation de terminal (putty/hyperterminal) 9600 bauds, 8, n, 1.



Connexion console sur un commutateur Cisco

[Source de l'image](#)

2. Password Recovery

À vrai dire, la procédure "password recovery", "recouvrement de mot de passe", ne permet pas de retrouver le mot de passe d'un routeur ou d'un commutateur Cisco.

Tout au plus, elle permet de reprendre la main sur un périphérique Cisco dont l'accès est limité par un mot de passe, de le changer ou de le lire si celui-ci n'est pas chiffré.

En laboratoire, cette procédure nous permettra de travailler sur des machines dont la configuration est vierge.

La procédure est bien distincte d'un commutateur à un routeur Cisco car il s'agit de plateformes différentes sur le plan matériel.

On peut reprendre la main sur un commutateur déjà configuré en interrompant son démarrage et en renommant le fichier de configuration initiale.

3. Password recovery sur un C2960

Source : http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960/software/release/15-0_2_se/configuration/guide/scg2960/swtrbl.html#53335

Etape 1 Connecter son PC à un logiciel d'émulation de terminal sur le port console du commutateur.

Etape 2 Fixer la vitesse à 9600 baud.

Etape 3 Eteindre le commutateur

Etape 4 Rallumer le commutateur en maintenant son doigt sur le bouton Mode jusqu'au moment où la LED System restera clignotante verte et puis deviendra brusquement ambre et puis verte de manière permanente.

Un message apparait sur la console du commutateur :

```
The system has been interrupted prior to initializing the flash file system. The following commands will initialize the flash file system, and finish loading the operating system software:
```

```
flash_init
load_helper
boot
```

Etape 5 Initialiser la mémoire Flash :

```
switch: flash_init
```

Etape 6 Chargement de l'environnement :

```
switch: load_helper
```

Etape 7 Contenu de la Flash :

```
switch: dir flash:
```

Le système de fichiers apparaît :

```
Directory of flash:
13 drwx 192 Mar 01 1993 22:30:48 c2960-lanbase-mz.122-25.FX
11 -rwx 5825 Mar 01 1993 22:31:59 config.text
18 -rwx 720 Mar 01 1993 02:21:30 vlan.dat
```

```
16128000 bytes total (10003456 bytes free)
```

Etape 8 Renommer le fichier de configuration en `config.text.old`. Ce fichier contient le mot de passe.

```
switch: rename flash:config.text flash:config.text.old
```

Etape 9 Redémarrage du système

```
switch: boot
```

Etape 10 La console vous invite à entrer dans le mode setup. Répondez N :

```
Continue with the configuration dialog? [yes/no]: N
```

Etape 11 Entrer en mode privilège :

```
Switch> enable
```

Etape 12 Renommer le fichier de configuration en son nom original

```
Switch# rename flash:config.text.old flash:config.text
```

Etape 13 Charger le fichier en mémoire vivie :

```
Switch# copy flash:config.text system:running-config
Source filename [config.text]?
Destination filename [running-config]?
```

Etape 14 Frapper “Entrée”, valider, pour confirmer.**Etape 15 On peut alors changer le mot de passe.**

```
Switch# configure terminal
Switch (config)# enable secret password
Switch (config)# exit
Switch#
Switch# copy running-config startup-config
Switch# reload
```

9. Connexion à un routeur Cisco

Ce chapitre explique comment se connecter à un routeur Cisco en console physique et comment reprendre la main sur un routeur dont a perdu le mot de passe via un *password recovery*. On y trouvera enfin un descriptif sur la notion de registre de configuration d'un routeur Cisco.

1. Connexion à un périphérique Cisco

1.1. Connexion série / console

- Câble inversé (roll-over) du port COM1 du PC au routeur sur le port console.
- Lancer un logiciel d'émulation de terminal (putty/hyperterminal) 9600 bauds, 8, n, 1.



Connexion console à un routeur Cisco

1.2. Password Recovery

À vrai dire, la procédure "password recovery", "recouvrement de mot de passe", ne permet pas de retrouver un mot de passe d'un routeur Cisco.

Tout au plus, elle permet de reprendre la main sur un routeur dont l'accès est limité par un mot de passe, de le changer ou de le lire si celui-ci n'est pas chiffré. En laboratoire, cette procédure nous permettra de travailler sur des machines dont la configuration est vierge.

La procédure est bien distincte pour un commutateur Cisco car il s'agit d'une plateforme différente sur le plan matériel.

2. Password Recovery du routeur

- Nécessite un accès physique.
- On distingue une procédure connue pour les plateformes "routeur" et une pour les plateformes "commutateur"
- Le principe consiste à redémarrer la machine sans charger la configuration qui contient les paramètres de restriction d'accès.

2.1. Principe

Si des mots de passe ont été définis par un administrateur, ils sont situés dans la “startup-configuration” (fichier de configuration initiale) qui elle-même sera copiée dans la “running-configuration” (fichier de configuration courante) à l’issue du démarrage du routeur.

En évitant la “startup-configuration”, on évite le passage obligé de mots de passe d’administration, on accède au mode de configuration avec tous les privilèges.

Éviter une configuration existante permet de démarrer la machine avec une configuration vierge. Tout au plus, peut-on consulter, supprimer, modifier ou charger la configuration de démarrage qui existe toujours en NVRAM. Les mots de passe devraient être chiffrés.

Pour ce faire, il faut changer les paramètres de démarrage du routeur afin d’interrompre son démarrage. Un accès physique au routeur est nécessaire. Dans la console, on utilisera la combinaison des touches CTRL et PAUSE (sous Windows) pour accéder au mode minimal du routeur qui est appelé “ROM Monitor Mode”. Il s’agit d’un mode de bas niveau exécuté en RAM et situé en ROM qui permet entre autres de changer les valeurs du registre de configuration, de télécharger un IOS à partir d’une connexion IP ou série, de configurer la vitesse de la console, etc.

Le changement s’opère sur la valeur du registre de démarrage. Une fois le routeur redémarré, il n’y aura plus de mot de passe demandé mais aussi plus aucune configuration. Notons aussi que les interfaces n’auront pas été démarrées. Il faudra explicitement les activer par un “no shutdown”. On pourra par contre changer n’importe quel paramètre de la machine. Enfin, il ne faudra pas oublier de changer la valeur du registre à partir du mode normal sur sa valeur par défaut. Autrement la “startup-configuration” sera toujours ignorée si le routeur redémarre pour une raison quelconque.

2.2. Conditions

La seule condition est d’avoir un accès aux ports console ou AUX et de pouvoir redémarrer le routeur.

Connexion à la console du routeur :

- Câble inversé (roll-over) du port COM1 du PC au routeur sur le port console.
- en USB
- Lancer un logiciel d’émulation de terminal (putty/hyperterminal) 9600 bauds

3. Procédure Password Recovery d’un routeur Cisco

Quelle que soit la plateforme routeur matériel, la procédure est toujours la même.

Toutefois, selon le routeur, les commandes en “ROM Monitor Mode” peuvent changer. Il suffit de les connaître ou de les trouver grâce à l’aide (commande help ou ?).

```
Press RETURN to get started!
```

```
Router>enable
Password:
Password:
```

Le routeur demande un mot de passe pour accéder au mode privilège.

3.1. Redémarrage du routeur et CTRL-PAUSE endéans les 60 secondes. Entrée dans le mode ROM Monitor


```
rommon1>
```

3.2. Modification de la valeur du registre de démarrage

```
rommon1>confreg 0x2142
```

3.3. Redémarrage du routeur (et esquive de la startup-configuration)

```
rommon2>reset
System Bootstrap, Version 12.1(3r)T2, RELEASE SOFTWARE (fc1)
Copyright (c) 2000 by cisco Systems, Inc.
cisco 2811 (MPC860) processor (revision 0x200) with 60416K/5120K bytes of memory

Self decompressing the image :
*##### [OK]
---Sortie omise---
--- System Configuration Dialog ---

Continue with configuration dialog? [yes/no]: no

Press RETURN to get started!
```

Tapez no pour éviter le mode setup

3.4. Visualisation de la configuration de démarrage

```
Router#show startup-configuration
Using 499 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
enable secret 5 $1$mERr$.TUcXMCAonRND.tFXUzor0
!
```

3.5. Vérification de la présence d'un fichier de configuration de démarrage

```
Router#cd nvram:
Router#dir
Directory of nvram:/

124  -rw-          0          <no date>  startup-config
125  ----          0          <no date>  private-config

129016 bytes total (128964 bytes free)
```

3.6. Modification des paramètres

```
Router>enable
Router#configure terminal
Router(config)#[...]
Router(config)^Z
Router#copy running-config startup-config
```

3.7. Rétablissement du registre de démarrage et redémarrage

```
Router#configure terminal
Router(config)#config-register 0x2102
Router(config)#exit
Router#reload
```

3.8. Vérification des paramètres du registre

```
Router#show version
Cisco IOS Software, 2800 Software (C2800NM-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE SOFTWARE (f\
c2)
---sortie ommise et à la dernière ligne :---
Configuration register is 0x2102
```

4. Registre de configuration

Un router Cisco dispose d'un code de configuration de registre sur une valeur de 16 bits. Cette valeur est stockée en NVRAM. On la configure via les commandes "confreg" en ROMMON ou "config-register" dans l'IOS.

4.1. Utilité

On peut l'utiliser pour configurer les différentes tâches :

- Forcer le routeur à entrer en ROMMON
- Sélectionner une source de démarrage ou un fichier de démarrage par défaut
- Activer ou désactiver la fonction Break
- Contrôler les adresses de Broadcast
- Charger un OS à partir de la ROM
- Récupérer des mots de passe
- Modifier la vitesse de ligne

4.2. Signification des 16 bits du registre

Bit	Fonction	Valeur	Hexa
15	Mode diagnostic et NVRAM ignorée	0x8---	1er Hexa
14	Broadcast IP n'a pas de numéros	0x4---	1er Hexa
13	Démarrage en ROM si erreur	0x2---	1er Hexa
12	Vitesse de Ligne	voir ci-dessous	1er Hexa
11	Vitesse de Ligne	voir ci-dessous	2ème Hexa
10	Broadcast IP tout à zéro	0x-4--	2ème Hexa
9	n/a	n/a	2ème Hexa
8	Break désactivé	0x-1--	2ème Hexa
7	Bit OEM activé	0x--8-	3ème Hexa
6	NVRAM ignorée	0x--4-	3ème Hexa
5	Vitesse de Ligne	voir ci-dessous	3ème Hexa
4	n/a	n/a	3ème Hexa
3	champ démarrage	voir ci-dessous	4ème Hexa
2	champ démarrage	voir ci-dessous	4ème Hexa
1	champ démarrage	voir ci-dessous	4ème Hexa
0	champ démarrage	voir ci-dessous	4ème Hexa

4.3. Le bit 13 sur le premier hexa

Ce champ permet au routeur de démarrer en ROM si tout autre démarrage est impossible sur les 4 premiers bit avec une valeur de 1 qui donne 0010 cela donne 0x2---. Il s'agit de la valeur par défaut.

4.4. Le bit 8 sur le deuxième hexa

Le bit 8 est à une valeur par défaut de 1 ce qui donne en hexa 0x-1--. Cette configuration ignore l'interruption du Break après les 60 secondes du démarrage.

4.5. Le bit 6 sur le troisième hexa

Ignorer la NVRAM, le bit 6 sur le troisième hexa

Si le bit est à 1, on a en hexa 0x--4-

Si le bit est à 0, on a en hexa 0x--0-

4.6. Les bits 3-2-1-0 sur le dernier hexa.

Ces bits sont appelés "boot fields".

Les valeurs classiques sont en hexa 0x---0, 0x---1 et 0x---2 à 0x---F

Valeur	Signification
0x---0	Reste en ROM Monitor mode
0x---1	Démarrage sur la première image disponible
0x---2 à 0x---F	Démarrage par défaut à partir de la Flash

Pour démarrer une image particulière, on peut utiliser la commande `boot system` :

```
(config)#boot system flash filename
```

ou

```
(config)#boot system rom
```

ou

```
(config)#boot system {rtp|tftp|ftp} filename [ip-address]
```

4.7. Les bits de vitesse (5-12-11)

La vitesse de la ligne console peut être définie sur les trois premiers hexa. Par défaut, la vitesse est définie à 9600 bauds :

Bauds	Bit 5	Bit 12	Bit 11
115200	1	1	1
57600	1	1	0
38400	1	0	1
19200	1	0	0
9600	0	0	0
4800	0	0	1
2400	0	1	1
1200	0	1	0

4.8. Contrôle des adresses de Broadcast (bits 10 et 14)

On peut déterminer le format des adresses de Broadcast via les bits 10 et 14.

Exemples

Démarrage par défaut en 9600 bauds en console (0x2102) :

Valeur Hexa	2	-	-	-	1	-	-	-	0	-	-	-	2	-	-	-
Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Valeur binaire	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0

Démarrage en évitant le contenu de la NVRAM (startup-config) (0x2142) :

Valeur Hexa	2	-	-	-	1	-	-	-	4	-	-	-	2	-	-	-
Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Valeur binaire	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0

10. Méthode Cisco IOS CLI

Ce document est une initiation à la méthode de configuration des commutateurs et des routeurs Cisco. On y évoque la hiérarchie des modes de configuration utilisateur, privilégié et de configuration. On y énonce les facilités de configuration, d'aide, de raccourcis clavier et différentes astuces.

1. Introduction

Un grand nombre de produits Cisco (routeurs, commutateurs, point d'accès, pare-feux, etc.) sont hautement configurables en ligne de commande (CLI) IOS. Le document "[Cisco IOS Internetwork Operating System](#)" évoque les différentes variantes du système d'exploitation Cisco Systems.

Quels sont les points à remarquer en tant qu'utilisateur d'un périphérique qui fonctionne en Cisco IOS ?

Une commande validée est immédiatement exécutée en mémoire vive. Les commandes répondent à une certaine logique qui généralement suit celle des protocoles mis en oeuvre.

Les tâches de diagnostic sont simples et explicites.

Une aide performante accompagne ce mode de configuration.

Certaines configurations peuvent comporter un grand nombre de lignes, mais sont susceptibles de subir des automatisations (exécution de scripts, traitement du texte, copier/coller en console, téléchargement de configuration en FTP ou TFTP, etc.). L'IOS évolue avec les besoins des clients qui demandent de configurations robustes et lisibles. Par exemple, les valeurs par défaut n'apparaissent pas explicitement dans la configuration. La lecture est simplifiée, mais la connaissance des protocoles et du produit devient cruciale. Il s'agit d'un très bon exercice des professionnels du domaine, car ce type d'environnement est souvent rencontré.

Il existe d'autres modes de configuration que la ligne de commande tels que par un navigateur Web ou un logiciel JAVA exécuté en local ou à distance tel que [Cisco Security Device Manager \(SDM\)](#). Mais ces facilités nécessitent la plupart du temps une configuration TCP/IP préalable qu'il faudra exécuter en ligne de commande. Ces modes de configuration établissent un tunnel TCP/IP sécurisé pour exécuter les commandes à distance grâce à un logiciel client et pour transférer des fichiers.

Ces méthodes de configuration et de gestion en ligne de commande sont identiques sur un routeur ou un commutateur faisant fonctionner l'IOS Cisco. On peut considérer que l'on est proche d'un environnement de type "shell" proche d'autres systèmes d'exploitation bien connus. On peut aussi considérer que le CLI Cisco IOS est une compétence encore très recherchée aujourd'hui.

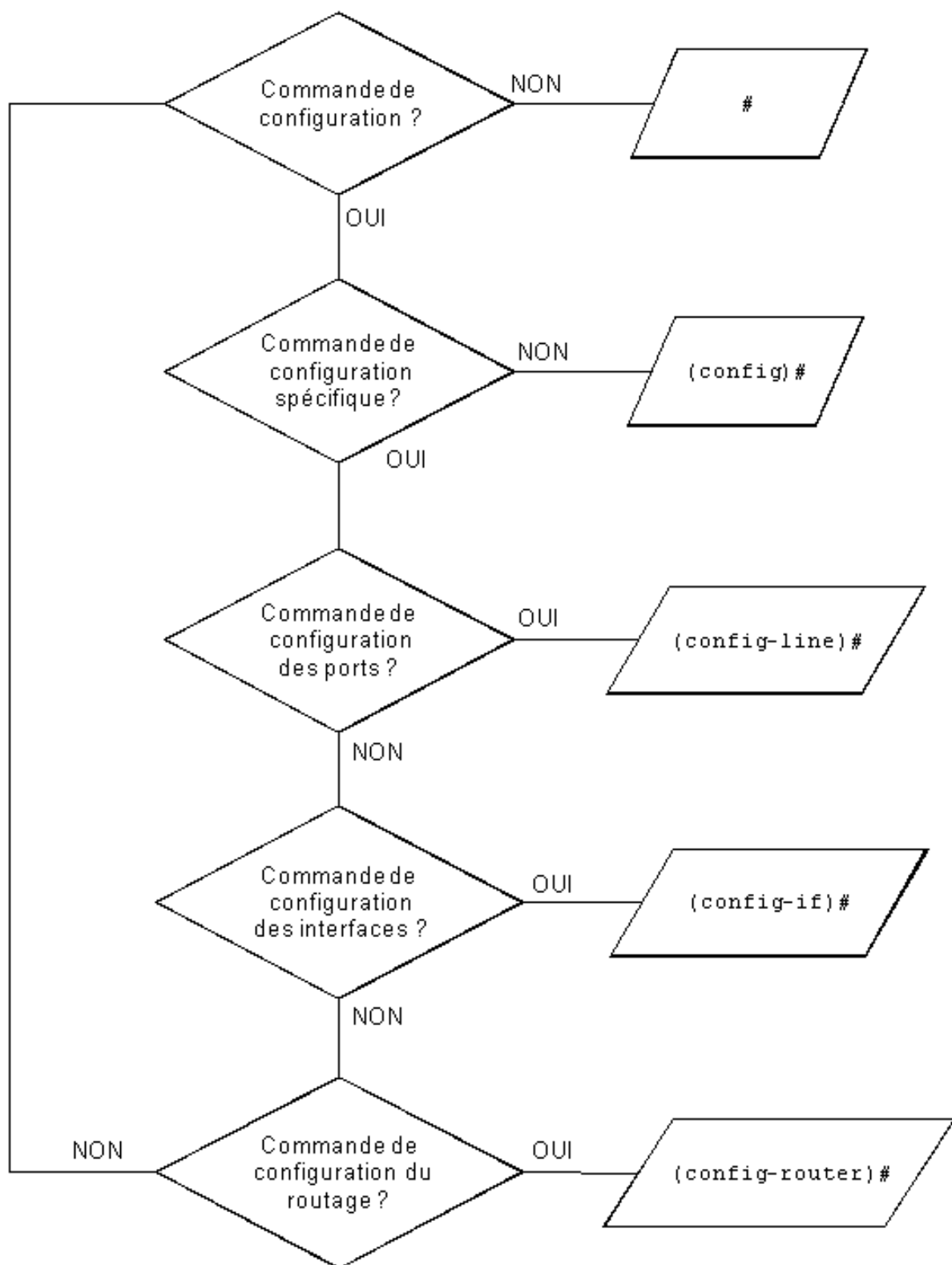
2. Hiérarchie CLI

On se base sur l'invite pour se situer dans un mode. Il faut les droits d'administration (mode privilège) pour exécuter des tâches de diagnostic ou de gestion. Toutes les commandes de configuration de la machine s'exécutent en mode de configuration globale. Toutes les commandes spécifiques pour les interfaces, le routage, des services (dhcp, nat, firewall, ...) ont leur mode configuration particulier.

On trouve au moins trois niveaux hiérarchiques CLI principaux :

1. Mode utilisateur (User EXEC mode) : session, diagnostic

- Mode privilège (Privileged EXEC mode) : Administration et gestion (commandes show, debug, copy, erase, ...)
- Mode configuration globale (Global Configuration mode)
- Modes de configuration spécifiques



Hiérarchie Command Line Interface (CLI) Cisco IOS

2.1. Passage en mode privilège

```
Switch>enable
Switch#
```

C'est le mode "root" du commutateur qui donne au plus au niveau de privilège sur le périphérique Cisco (level 15). Dans ce mode aucune commande de configuration n'est admise.

Il supporte uniquement des commandes :

- d'information : debug, show, telnet, ssh, ...
- de diagnostic : ping, ...
- de gestion : telnet, ssh, erase, copy, ...

2.2. Passage en mode de configuration globale

Pour configurer le commutateur, il est nécessaire de passer en configuration globale. Ce mode ne supporte aucunes des commandes du mode privilège.

```
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

2.3. Configuration d'une interface

```
Switch(config)#interface G0/0
Switch(config-if)#
```

2.4. Passage aux modes inférieurs

```
Switch(config-if)#exit
Switch(config)#exit
Switch#
```

3. Aide, autocomplétion, historique, raccourcis, confort

3.1. Aide

Une aide est accessible via le point d'interrogation ?.

```
Switch#show ?
aaa                Show AAA values
access-expression  List access expression
access-lists       List access lists
acircuit           Access circuit info
adjacency          Adjacent nodes
aliases            Display alias commands
alps               Alps information
ancp               ANCP information
apollo             Apollo network information
appletalk          AppleTalk information
application        Application Routing
```

```

arap          Show Appletalk Remote Access statistics
archive       Archive functions
arp           ARP table
async         Information on terminal lines used as router interfaces
authentication Shows Auth Manager stats, registrations or sessions
auto          Show Automation Template
backup        Backup status
banner        Display banner information
beep          Show BEEP information
bfd           BFD protocol info
bgp           BGP information
--More--

```

```

Switch#show vlan ?
access-log    VACL Logging
access-map    Vlan access-map
brief         VTP all VLAN status in brief
dot1q         Display dot1q parameters
filter        VLAN filter information
group         VLAN group(s) information
id            VTP VLAN status by VLAN id
ifindex       SNMP ifIndex
internal      VLAN internal usage
mtu           VLAN MTU information
name          VTP VLAN status by VLAN name
private-vlan  Private VLAN information
remote-span   Remote SPAN VLANs
summary       VLAN summary information
|            Output modifiers
<cr>

```

```

Switch#show vlan name ?
WORD  A VTP VLAN name

```

3.2. Autocomplétion

Les commandes s'autocomplètent avec la touche de tabulation.

#sh <tabulation> devient #show

```

switch#sh
switch#show

```

3.3. Signalement d'erreurs

L'environnement indique l'endroit d'une erreur.

```

Switch#show cown
          ^
% Invalid input detected at '^' marker.

```

3.4. Commandes abrégées

Les commandes s'abrègent s'il n'y a pas d'ambiguïté.


```
Switch#conf t
```

au lieu de `#configure terminal`

```
Switch#sh ip int b
```

au lieu de `show ip interface brief`

```
Switch#copy r s
```

au lieu de `copy run start` ou `copy running-config startup-config`

Sans confirmation :

```
Switch#wr
```

au lieu de `write memory`

Dernier exemple sur les interfaces

```
Switch(config)#int g0/0
```

au lieu de `(config)#interface GigabitEthernet 0/0`

3.5. Ambiguïté

- En cas d'ambiguïté, l'environnement propose les choix.

```
Switch#con t
% Ambiguous command: "con t"
Switch#con
Switch#con
Switch#con?
configure connect

Switch#con
```

3.6. Raccourcis clavier

Raccourcis clavier ([Emacs](#)) : on peut faire défiler l'historique des commandes avec les flèches du haut et du bas, on peut revenir au mode privilège directement (CTRL-Z), etc.

Séquence	Description
CTRL-A	Début de ligne
CTRL-E	Fin de ligne
CTRL-P ou bas	Commande précédente
CTRL-N ou haut	Commande suivante
CTRL-F ou droite	Curseur vers la droite
CTRL-B ou gauche	Curseur vers la gauche
CTRL-Z	Retour au mode privilège
CTRL-C	Interruption
CTRL-SHIFT-6 / CTRL-SHIFT\$	Interruption forcée
TAB	Complète un commande
Backspace	Efface un caractère vers la gauche
CTRL-R	Réaffichage de la ligne
CTRL-U	Efface la ligne
CTRL-W	Efface un mot
ESC-b	recule d'un mot
ESC-f	Avance d'un mot

4. Facilités à configurer

4.1. Commande do

La commande do permet d'exécuter une commande du mode privilège dans un autre mode.

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#show run
      ^
% Invalid input detected at '^' marker.

Switch(config)#do show run
Building configuration...
```

4.2. Apparition des logs

Par défaut les logs apparaissent dans la console, mais pas en terminal distant.

Dans une session Telnet ou SSH, les logs apparaissent suite à la commande :

```
Switch#terminal monitor
```

4.3. Messages synchronisés de la console

On peut recevoir des messages du routeur qui trouble l'entrée des commandes. Voici une illustration :

```
Switch#confi
%SYS-5-CONFIG_I : Configured from console by console
Enter configuration commands, one per line. End with CNTL/Z.
Switch#
```

Pour éviter ce problème, on peut demander un logging synchronisé sur la console :

```
Switch(config)#line con 0
Switch(config-line)#logging synchronous
```

Voici l'éventuel résultat :

```
Switch#confi
%SYS-5-CONFIG_I : Configured from console by console
Switch#configure
```

4.4. Désactivation des recherches DNS

Sans résolution DNS, une frappe erronée ne correspondant pas à une commande peut être interprétée en mode privilège comme une tentative de connexion distante SSH et bloquer un certain temps la console :

```
Switch#zozo
```

On désactive la recherche DNS avec la commande :

```
Switch(config)# no ip domain-lookup
```

ou

```
Switch(config)# no ip domain lookup
```

4.5. Filtrer les sorties des commandes `show` et `more`

La commande `show` offre énormément d'informations sur un système Cisco. La commande `more` permet d'afficher un fichier à la manière de la commande Unix.

A condition de bien connaître les sorties et de savoir ce que l'on cherche précisément, Cisco IOS propose une fonctionnalité de traitement des sorties à la manière des redirections (*pipe*) Bash ou PowerShell. Le tube (*pipe*) redirige la sortie pour être traitée par un programme `begin`, `include` ou encore `count` parmi d'autres.

```
*show running-config | ?
```

```
append      Append redirected output to URL (URLs supporting append operation
              only)
begin        Begin with the line that matches
count        Count number of lines which match regexp
exclude      Exclude lines that match
format       Format the output using the specified spec file
include      Include lines that match
redirect     Redirect output to URL
section      Filter a section of output
tee          Copy output to URL
```

Ces programmes ressemblent fort à des programmes Unix qui attendent un motif comme critère de filtrage. Ce motif est exprimé sous forme d'*expression rationnelle (RegExp, Regular Expression)*.

```
*show running-config | begin ?
```

```
LINE Regular Expression
```

Par exemple, pour afficher le statut des interfaces uniquement sur GigabitEthernet0/0

```
*show ip interface brief | include GigabitEthernet0/0
```

```
GigabitEthernet0/0      unassigned      YES unset  up
```

Par exemple, pour commencer une sortie sur un motif.

```
*show running-config | begin interface GigabitEthernet1/0
```

```
interface GigabitEthernet1/0
 media-type rj45
 negotiation auto
 spanning-tree portfast edge
!
interface GigabitEthernet1/1
 media-type rj45
 negotiation auto
 spanning-tree portfast edge
!
interface GigabitEthernet1/2
 media-type rj45
 negotiation auto
 spanning-tree portfast edge
```

```
!  
interface GigabitEthernet1/3  
  media-type rj45  
  negotiation auto  
  spanning-tree portfast edge  
!  
interface GigabitEthernet2/0  
  media-type rj45  
  negotiation auto  
--More--
```

Troisième partie Protocole IPv4

Tous les ordinateurs qui se connectent à ce qu'on appelle communément l'Internet sont identifiés de manière certaine par une adresse Internet (IP). Cette partie s'intéresse à la couche Internet en général, aux adresses IPv4 et aux masques de sous-réseau, au NAT, aux protocoles ICMP, ARP, UDP et TCP.

A la fin de la partie, à titre de diagnostic, on proposera plusieurs commandes de prise d'information et de l'observation de trafic TCP/IP.

La couche Internet est celle qui permet à deux ordinateurs situés à n'importe quel endroit du monde de communiquer directement entre eux. Les routeurs utilisent l'adressage du protocole IPv4 ou celui du protocole IPv6 pour acheminer les paquets jusqu'à leur destination. La gestion des adresses IP est confiée à des organismes régionaux (les RIRs). Actuellement le protocole le plus utilisé est IPv4. IPv6 et le NAT (la traduction d'adresses) sont des solutions à l'épuisement des adresses IPv4.

Même si la certification Cisco CCNA n'exige pas la connaissance des en-têtes IPv4 ou IPv6 (ni d'aucun autre protocole), il est utile de distinguer les différences entre ces en-têtes et de les comparer d'un protocole à l'autre. Par contre, il est unimaginable de se présenter à un examen Cisco ou à un entretien d'embauche dans le domaine des réseaux sans maîtriser l'adressage IPv4 et ses mécanismes de découpage.

Enfin, IPv4 est aidé par deux protocoles pour la résolution d'adresses et le contrôle : ARP et ICMP au niveau de la couche 3 (L3). Enfin, les protocoles TCP et UDP de la couche Transport des modèles OSI et TCP/IP, vus de manière comparative, font partie des sujets vérifiés dans la certification CCNA.

11. La couche Internet du modèle TCP/IP

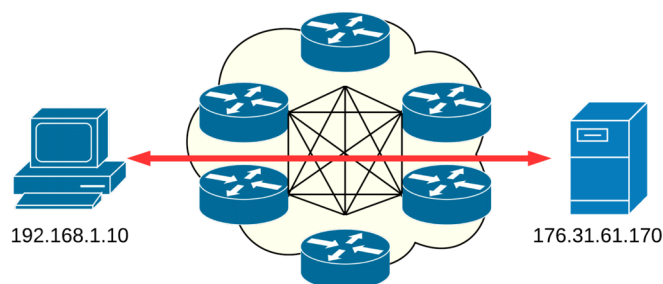
La couche Internet est celle qui permet à deux ordinateurs situés à n'importe quel endroit du monde de communiquer directement entre eux. Les routeurs utilisent l'adressage du protocole IPv4 ou celui du protocole IPv6 pour acheminer les paquets jusqu'à leur destination. La gestion des adresses IP est confiée à des organismes régionaux (les RIRs). Actuellement le protocole le plus utilisé est IPv4. IPv6 et le NAT (la traduction d'adresses) sont des solutions à l'épuisement des adresses IPv4.

1. Définition de la couche Internet

1. La couche Internet est celle qui s'occupe d'adresser globalement les interfaces : elle remplit une fonction d'**adressage**.
2. Elle détermine les meilleurs chemins à travers les inter-réseaux : elle remplit une fonction de **routing**.

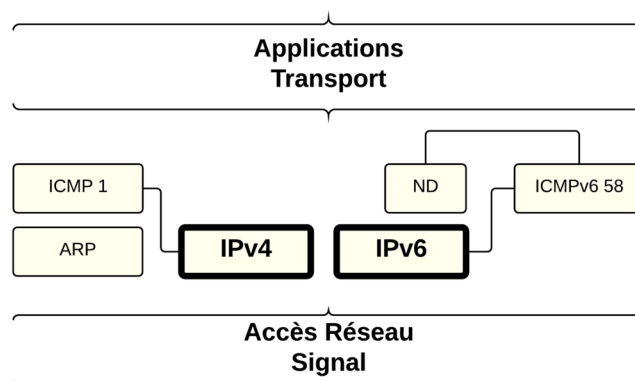
Elle utilise un des protocoles Internet (IPv4 ou IPv6). Ceux-ci ont pour caractéristique communes de fonctionner en mode **non fiable** et en mode **non connecté**.

En bref, la couche Internet est celle qui permet à ces deux ordinateurs de communiquer directement entre eux via un inter-réseau probablement à l'échelle du globe : à travers l'Internet.



La couche Internet est celle qui permet à ces deux ordinateurs de communiquer directement entre eux via un interréseau.

2. Modèle TCP/IP : couche Internet



Modèle TCP/IP : couche Internet

3. Protocoles de couche Internet

IPv4 et IPv6 sont accompagnés d'autres protocoles comme ARP, ICMP et ICMPv6 :

- La couche Internet remplit aussi le rôle de **résolution d'adresses** : ARP ([RFC826](#)) en IPv4 et **Neighbor Discovery (ND)** ([RFC4861](#)) en IPv6.
- IPv4 dispose d'**ICMP** ([RFC792](#)) et IPv6 d'**ICMPv6** ([RFC443](#)) pour du **diagnostic et des messages d'erreurs**.
- IPv4 et IPv6 sont aidés par des **protocoles de routage** pour maintenir le routage Internet (BGP, OSPF, EIGRP)

Avec ICMPv6 qui embarque des fonctions de **configuration du réseau**, d'**autoconfiguration des interfaces** et de **maintenance de relations de voisinage**, soit l'**autoconfiguration automatique sans état** (Stateless Address Autoconfiguration) ([RFC4862](#)), la couche 3 (L3, Internet/Réseau) maîtrise nativement la gestion de cet adressage de 128 bits. Un marché s'ouvre pour exploiter les nouvelles fonctionnalités d'IPv6 en termes de gestion (IPAM, IP Address Management).

4. Protocoles Internet

Quelle que soit leur version, IPv4 ou IPv6, les "*Internet Protocols*" répondent à quelques principes majeurs :

- d'une **communication de bout en bout** : Les adresses d'origine et de destination utilisées pour adresser les machines communicantes sont joignables de bout en bout
- du **meilleur effort** : Les paquets sont acheminés sans garantie quant à leur acheminement, Méthode de qualité de service (QoS) par défaut.
- Il est aussi **réputé "non-fiable"** : **Sans mécanisme de fiabilité** (pas de contrôle de flux, pas d'accusés de réception, pas gestion des erreurs, il est néanmoins robuste)
- Il est **"non orienté connexion"** : Un protocole "orienté connexion" est celui qui établit, maintient et termine une communication.
- **Unicité des adresses** : les adresses Unicast sont censées être uniques dans un Interréseau. Les adresses Multicast sont ces adresses uniques qui peuvent être adressées à plusieurs noeuds, même à travers un interréseau (IPv6).

Par ailleurs, les protocoles IP présentent d'autres caractéristiques :

- IP fait le **lien ENTRE l'infrastructure qui transporte les données ET les services offerts**. Il est donc **central et critique**.
- Le NAT (NAT44, NAT66, NAT444, CG-NAT, NAT64, tous types de NAT) **contrevient au principe d'une communication de bout en bout** empêchant d'exploiter pleinement le potentiel de TCP/IP. Si le NAT est indispensable dans l'exploitation des réseaux IPv4, on envisagera toute solution NAT avec mesure. On privilégiera des solutions et des architectures basées sur du matériel et des protocoles de sécurité et "applicatifs" comme des *proxys*.
- Des fonctionnalités/modèles de **Qualité de Service (QoS)** autres que le "meilleur effort" peuvent être mises en oeuvre.
- Selon le service utilisé, **TCP** prend en charge les fonctionnalités de fiabilité.
- Sur le plan physique, **la couche sous-jacente (Accès Réseau)** peut éventuellement aussi prendre en charge des mécanismes de fiabilité.

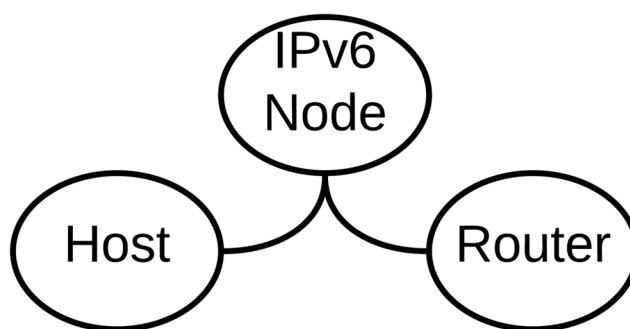
Sur le plan fondamental, un “*Internet Protocol*” est un protocole qui permet :

- D’acheminer des paquets de données d’une extrémité à l’autre de l’interréseau,
- à travers des réseaux distincts (donc différents) interconnectés entre eux.

Les routeurs transfèrent le trafic d’une origine à une destination et prennent leurs décisions sur base des adresses IP contenues dans les paquets.

5. Internet : rôles

Au niveau de la couche 3 (L3), pour les protocoles IP, les hôtes TCP/IP sont reconnus comme des “noeuds” (*nodes*) qui peuvent prendre un des deux rôles : hôte terminal (*end host*) ou routeur (*router*).



IPv6 voit deux rôles : hôte terminal et routeur.

- Les hôtes terminaux qui disposent de une ou plusieurs interfaces attachées à un lien.
- Les routeurs qui disposent de plusieurs interfaces attachées à des liens et qui transfèrent le trafic qui ne leur est pas destiné. Ils prennent leurs décisions sur base de leur table de routage. Le routeur examine les en-têtes IP au niveau du champ d’adresse de destination pour prendre ses décisions de routage. Il filtre (il ne transfère pas) le Broadcast.

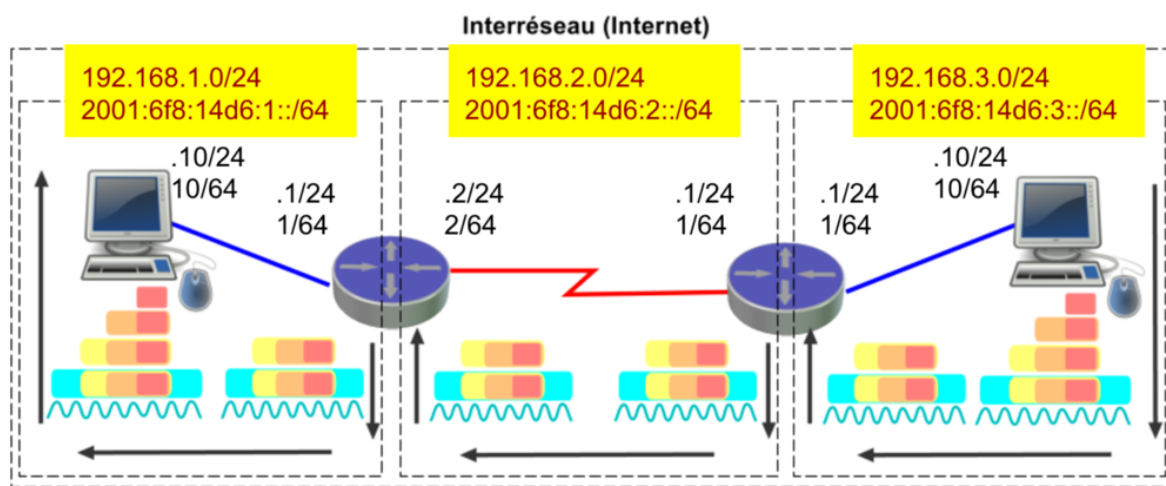
6. Routeur IP

- Seuls les routeurs sont capables de transférer les paquets d’une interface à une autre.
- Un routeur transfère le trafic qui arrive sur l’une de ses interfaces en fonction de l’adresse IP destination trouvée dans le paquet ; précisément il compare cette destination à une entrée de sa table de routage qui dispose de toutes les destinations d’un inter-réseau.
- Un routeur transfère le trafic qui ne lui est pas destiné, par définition.
- Un routeur limite les domaines de Broadcast sur chacune de ses interfaces.
- Un routeur échange avec ses voisins des informations concernant les différentes destinations (des réseaux à joindre) grâce à des protocoles de routage.

De plus, de manière obligatoire en IPv6, les routeurs configurent le réseau.

7. Routage entre domaines IP

- Deux noeuds IP (hôtes terminaux, interfaces, cartes réseau, PC, smartphone, etc.) doivent appartenir au même réseau, au même domaine IP pour communiquer directement entre eux.
- Quand les noeuds IP sont distants, ils ont besoin de livrer physiquement (L2) leur trafic à une passerelle, soit un routeur.
- D'une extrémité à l'autre de l'inter-réseau, les adresses IP ne sont pas censées être modifiées (sauf NAT) par les routeurs. Par contre, le paquet est désencapsulé /réencapsulé différemment au niveau de la couche Accès (L2) au passage de chaque routeur.



Cheminement d'un paquet d'une extrémité à l'autre d'un inter-réseau.

8. Organisation des adresses IP

IPv4 offre un espace d'adressage de 32 bits, soit un espace de 2^{32} adresses, aujourd'hui épuisé.

IPv6 offre quant à lui un emplacement de 128 bits pour l'adressage, soit un espace de 2^{128} adresses.

Ce sont des adresses organisées de manière hiérarchique sur base géographique (globe/continent/pays/Fournisseur d'Accès Internet/Client).

Ces attributions d'adresses s'organisent comme suit :

1. La gestion de l'espace d'adresses est confiée par l'IANA aux différents RIRs (*Regional Internet Registries*) comme le RIPE-NCC, l'APNIC, ...
2. Les RIRs confient des blocs à des LIRs (*Local Internet Registries*), des ISP (FAI, Fournisseur d'Accès Internet) ou des grandes entreprises.
3. Les ISP (FAI) offrent des services de connectivité à leurs clients finaux (EU, *End Users*).

NB : Toute demande d'adresse IP doit être justifiée par un projet et une continuité.



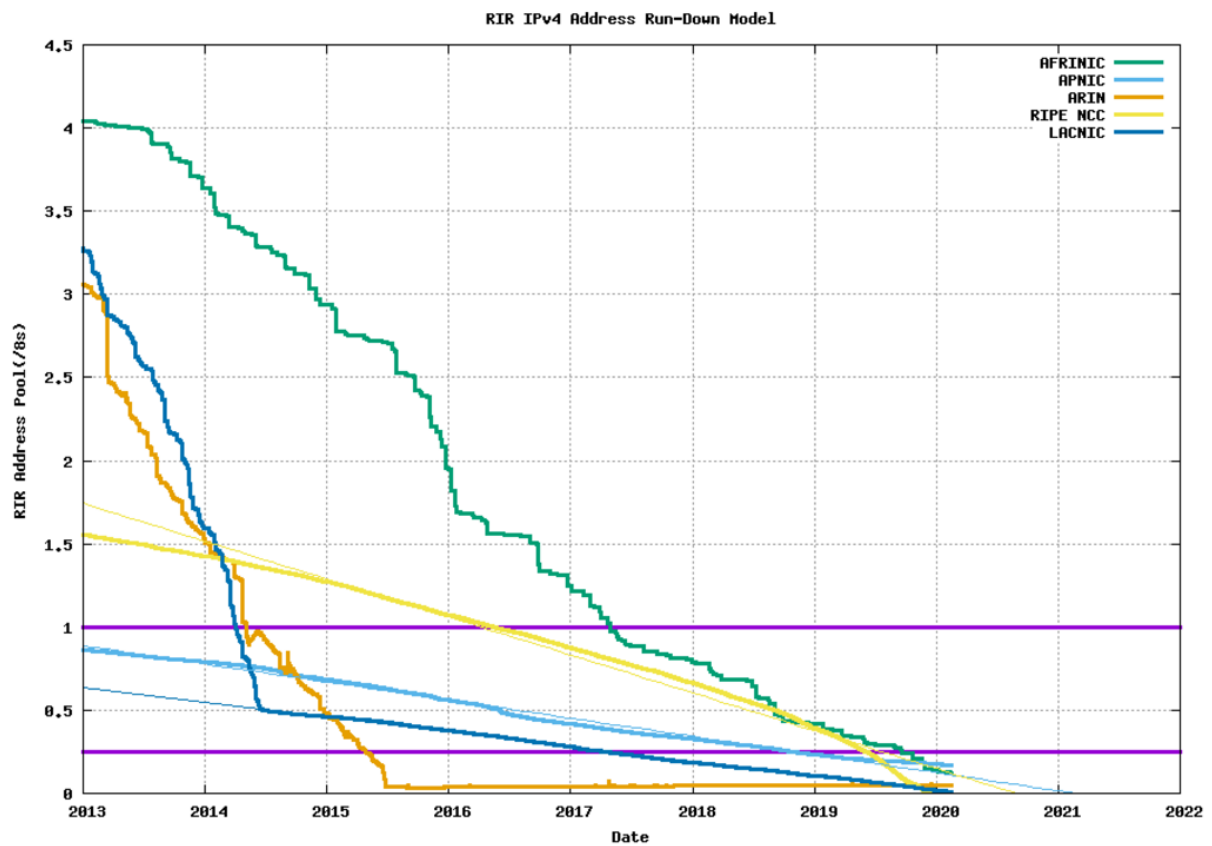
[RIRs (Regional Internet Registries)](<https://www.apnic.net/about-apnic/organization/history-of-apnic/history-of-the-regional-internet-registries/>)

9. Épuisement des adresses IPv4

Pour l'instant, la grande majorité des ressources Internet sont disponibles en IPv4. L'épuisement des adresses IPv4 (publiques) est l'épuisement du "pool" d'adresses IPv4 non allouées par les RIRs. Étant donné qu'il y a moins de 4,3 milliards d'adresses disponibles, l'épuisement a été anticipé depuis la fin des années 1980, alors que l'Internet a connu une croissance spectaculaire dès sa commercialisation. Cet épuisement est l'une des raisons du développement et du déploiement de son protocole successeur IPv6.

IPv6 est proposé et est déjà déployé dans les réseaux des opérateurs et des grands fournisseurs de contenu. Il se déploie progressivement dans les centres de données et sur les connexions domestiques. Toutefois peu d'entreprises le déploient dans leur LAN malgré sa présence *de facto*¹.

1. IPv6 est activé et fonctionne par défaut dans toute solution d'entreprise Microsoft Active Directory.

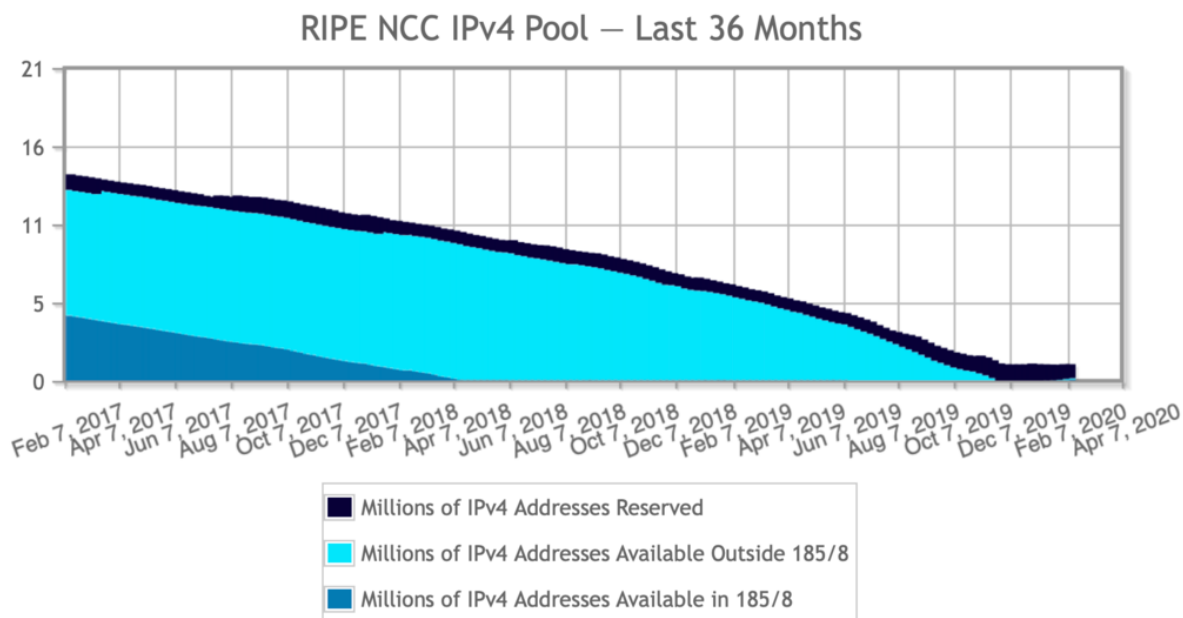


[Projection de Geoff Huston sur l'évolution des pools d'adresses IP par RIR](<https://ipv4.potaroo.net/>)

Source de l'image : Projection de Geoff Huston sur l'évolution des pools d'adresses IP par RIR

Le 17 avril 2018, le RIPE-NCC vient d'attribuer son dernier bloc IPv4 185/8 en blocs /22 seulement aux LIRs disposant déjà de blocs IPv4/IPv6. En conséquence :

- Les nouveaux entrants sont exclus d'IPv4.
- Trafic illégal d'adresses IPv4 prévisible.
- Cela signifie concrètement le début d'une impossibilité à déployer largement des services TCP/IPv4 au niveau global.



[RIPE NCC IPv4 Available Pool](<https://www.ripe.net/publications/ipv6-info-centre/about-ipv6/ipv4-exhaustion/ipv4-available-pool-graph>)

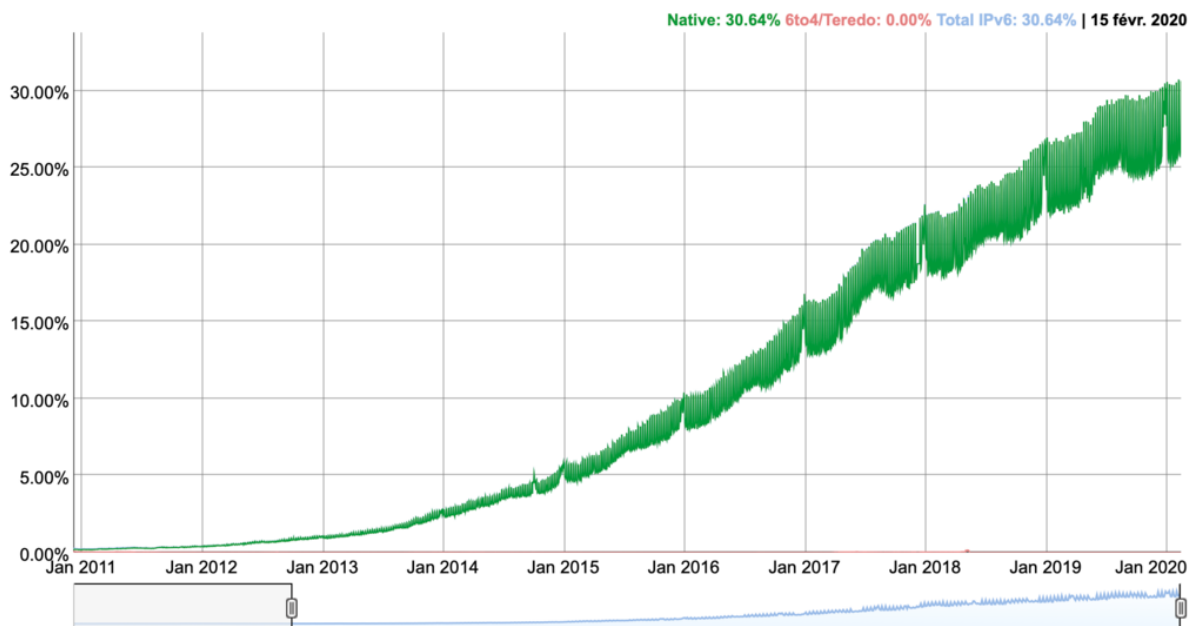
D'ici mai 2020, le RIPE-NCC pourra encore attribuer 9,3 millions d'adresses qui étaient réservées et qui ont été récupérées.

10. Transition vers IPv6

Étant donné que toutes les attributions d'adresse IPv4 sont épuisées à terme, IPv6 doit être déployé. Mais quel est le taux d'adoption d'IPv6 dans le Monde ?

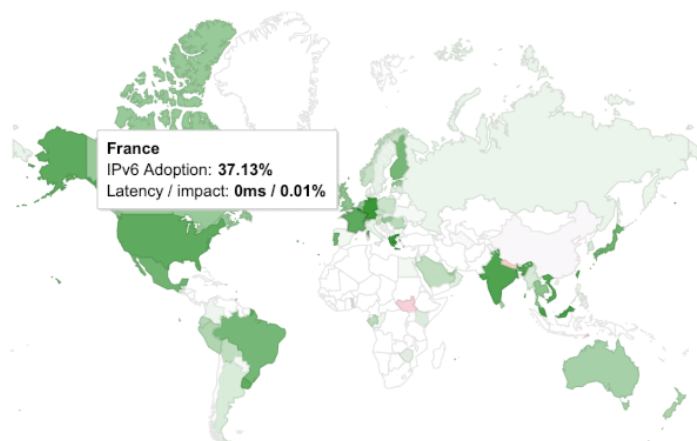
IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.



[IPv6 Adoption sur base des statistiques de Google](<https://www.google.com/intl/en/ipv6/statistics.html>)

Per-Country IPv6 adoption



[World](#) | [Africa](#) | [Asia](#) | [Europe](#) | [Oceania](#) | [North America](#) | [Central America](#) | [Caribbean](#) | [South America](#)

The chart above shows the availability of IPv6 connectivity around the world.

- Regions where IPv6 is more widely deployed (the darker the green, the greater the deployment) and users experience infrequent issues connecting to IPv6-enabled websites.
- Regions where IPv6 is more widely deployed but users still experience significant reliability or latency issues connecting to IPv6-enabled websites.
- Regions where IPv6 is not widely deployed and users experience significant reliability or latency issues connecting to IPv6-enabled websites.

[Adoption IPv6 par pays selon Google](<https://www.google.com/intl/en/ipv6/statistics.html#tab=per-country-ipv6-adoption&tab=per-country-ipv6-adoption>)

Vu que les hôtes terminaux ne peuvent utiliser que l'un ou l'autre des deux protocoles IP, on peut considérer que l'Internet IPv6 est un second Internet dont l'architecture va progressivement supplanter IPv4.

Cette phase de transition “duale” pourrait durer jusqu'à 10 ans et plus. Stéphane Bortzmeyer écrivait ceci en première phrase de son article du 7 juin 2018 intitulé “[Aurait-il fallu faire IPv6 « compatible » avec IPv4 ?](#)” : *Le déploiement du protocole IPv6 continue, mais à un rythme très réduit par rapport à ce qui était prévu et espéré. À cette vitesse, on aura encore de l'IPv4 dans 20 ou 30 ans.*

11. Généalogie IPv4/IPv6

D'un point de vue généalogique, IPv6 est un “vieux” protocole qui entre seulement maintenant dans sa phase de déploiement global. En réalité, la pénurie d'adresses a été prévue dès la mise en oeuvre commerciale d'IPv4 à la fin du XXème siècle. Au même moment, un nouveau protocole IPv6 est conçu. Entretemps, il fait l'objet de nombreuses expérimentations locales et globales.

Période	Protocoles Internet
1981	IPv4 Classes d'adresses (RFC791).
1985	Masques de sous-réseaux (RFC950).
1993	CIDR-VLSM (RFC4632).
1994	NAT (RFC1631 rendu obsolète par RFC3022).
1996	Adressage privé (RFC1918).
1995-1998	IPv6 (RFC2460).
World IPv6 Launch Day (2012)	La plupart des fournisseur d'accès Internet, des fabricants de matériel réseau grand public et des entreprises Web à travers le monde ont activé de manière permanente le protocole IPv6 pour leurs produits et service.
2017	RFC8200 : IPv6 passe de statut “DRAFT STANDARD” à “INTERNET STANDARD”.

Depuis 2012, nous sommes entrés dans une longue période de transition de la **double pile IPv4/IPv6** qui pourrait durer plus de 10 ans.

Il n'est plus envisagé de manière crédible de traduire le nouveau protocole dans l'ancien, IPv6 dans IPv4, autrement que pour dépanner ou bricoler. Par contre, l'inverse, c'est-à-dire traduire l'ancien protocole dans le nouveau,

IPv4 dans IPv6, annonce la prochaine étape de transition. Les solutions NAT64/DNS64 offrent la possibilité d'un déploiement local "IPv6 Only" tout en rendant encore accessible certaines ressources "seulement IPv4". Ce scénario n'est pas encore globalement d'actualité même si le cas a déjà fait l'objet d'expérimentations convaincantes². Une solution NAT64/DNS64 est encore certainement aujourd'hui une solution coûteuse pour une entreprise et relèverait plutôt d'une responsabilité d'opérateur³ et de gestionnaires de centres de données.

12. Nouveautés IPv6

D'un point de vue strictement technique, on peut considérer qu'IPv6 est le résultat amélioré de notre longue expérience d'IPv4. D'un point de vue comparatif, IPv4 et IPv6 sont à la fois si semblables et tellement différents. Quelles sont les nouveautés en IPv6 par rapport à IPv4 ?

- Adressage incommensurablement étendu sur 128 bits
- Le Broadcast disparaît au profit du Multicast
- Plus besoin de NAT, *a priori*
- ARP disparaît au profit de Neighbor Discovery (ICMPv6)
- Entrée DNS IPv6 AAAA
- **Le routeur configure le réseau**
- Adressage automatique local obligatoire
- Autoconfiguration automatique sans état seulement, DHCPv6 seulement ou les deux pour autant de préfixes à distribuer.
- Plug-and-Play par défaut
- DHCPv6-PD, solutions IPAM
- Reprise en main de la sécurité des accès, des règles de filtrage, de l'architecture sécurisée

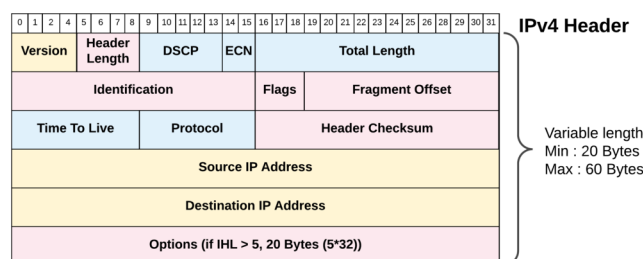
2. [IPv6-only at Microsoft](#) : "Hopefully, migrating to IPv6 (dual-stack) is uncontroversial at this stage. But for us, moving to IPv6-only as soon as possible solves our problems with IPv4 depletion and address oversubscription. But it also moves us to a simpler world of network operations where we can concentrate on innovation and providing network services, instead of wasting energy battling with such a fundamental resource as addressing."

3. [RFC7269](#).

12. En-têtes IPv4 et IPv6

Même si la certification Cisco CCNA n'exige pas la connaissance des en-têtes IPv4 ou IPv6 (ni d'aucun autre protocole), il est utile de distinguer les différences entre ces en-têtes et de les comparer d'un protocole à l'autre.

1. En-tête IPv4



En-tête IPv4

On retiendra qu'un en-tête IPv4 dispose d'une taille variable de minimum 20 octets et de maximum 60 octets. La taille d'un paquet entier peut aller jusqu'à 65536 octets. Mais certaines technologies L2 (de couche 2) comme Ethernet ne supportent que des paquets d'une taille de 64 à 1500 octets (*a contrario*, FDDI supporte des paquets de maximum 4478 octets, Frame-Relay supporte des paquets d'une taille variant entre 46 et 4470 octets).

En IPv4, quand le paquet doit passer par un chemin qui connaît une liaison dont le MTU (Maximum Transfer Unit) est inférieur à la taille du paquet, le routeur qui connecte cette liaison réalise une fragmentation du paquet original, divisant la charge dans la taille adaptée en répétant l'en-tête. Des champs de fragmentation identifient les paquets qui sont reconstruits uniquement à l'endroit de la destination finale. Autrement dit, la fragmentation ne peut s'opérer qu'une seule fois dans la communication.

On reconnaîtra aussi le champ "Time To Live", qui est codé sur 8 bits (de 0 à 255) dont la valeur démarre au maximum. La valeur de ce champ est dés-incrémentée de 1 au passage de chaque routeur. Quand cette valeur atteint 1, le paquet ne peut plus être transféré et l'interface du routeur qui a reçu le paquet retourne à la source un message ICMP "Time Exceeded".

Le champ "Protocol" annonce un protocole de couche supérieure (L4, Transport) embarqué dans la charge du paquet. Les valeurs les plus courantes sont les suivantes.

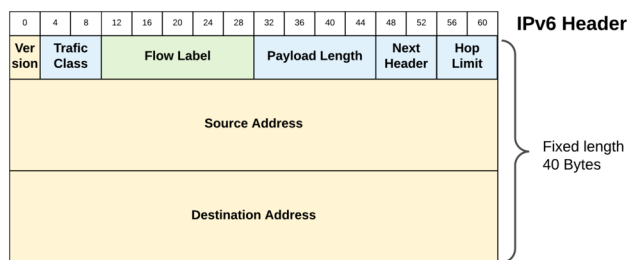
Valeur	Protocol
1	ICMP ¹
6	TCP
17	UDP
41	IPv6 ²
47	GREs
50	ESP (IPSEC)
51	AH (IPSEC)
58	ICMPv6 ³
88	EIGRP
89	OSPF

1. Uniquement pour IPv4.

2. Le code protocol IPv4 41 identifie un tunnel 6in4 qui embarque de l'IPv6 dans de l'IPv4.

3. Uniquement pour IPv6.

2. En-tête IPv6 de base



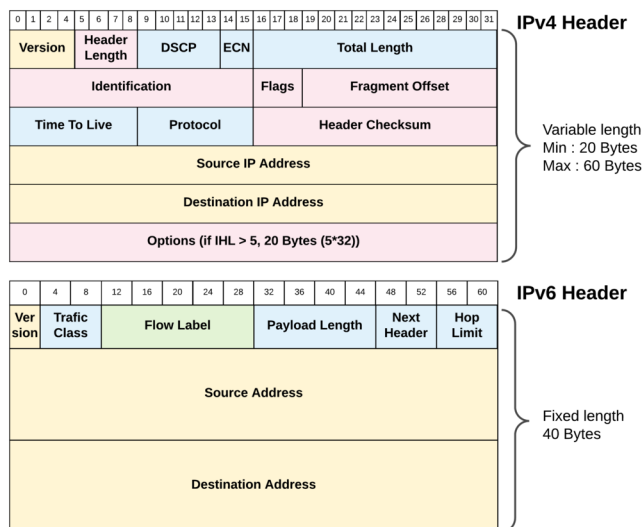
En-tête IPv6

Par rapport à un en-tête IPv4, IPv6 vise à minimiser la surcharge à son niveau et à simplifier le processus de traitement des paquets sur les routeurs.

A première vue, un en-tête IPv6 a été simplifié pour laisser principalement quelques champs de taille fixe comme les adresses source et destination codées sur 128 bits, un champ de durée de vie (Hop Limit) et celui qui annonce une charge supérieure (Next Header).

Comparativement à IPv4, un en-tête IPv6 dispose des caractéristiques suivantes.

- Un en-tête IPv6 de taille fixe de 40 octets.
- Disparition du champ “Header Checksum”, IHL.
- La fonction de fragmentation a été retirée des routeurs et disparaissent de l’en-tête de base pour être reportées dans des en-têtes d’extension.
- Les champs “Options” remplacées par des en-têtes d’“Extensions”.
- Les champs d’adresses sont des mots de 128 bits.
- Le champ IPv6 “Next Header” correspond au champ “Protocol” IPv4.
- Le champ IPv6 “Hop Limit” correspond au champ “Time to Live” IPv4.
- Le champ IPv6 “Flow Label” est nouveau



Comparaison des en-têtes IPv4 et IPv6

3. En-têtes IPv6 d'extension

IPv6 encapsule tout le trafic dans un en-tête fixe de base constitué de huit champs.

Les extensions d'IPv6 peuvent être vues comme un prolongement de l'encapsulation d'IPv6 dans IPv6.

À part l'extension de "proche-en-proche" traitée par tous les routeurs intermédiaires, les autres extensions ne sont prises en compte que par les équipements destinataires du paquet.

Une extension a une longueur multiple de 8 octets. Elle commence par un champ "Next Header" d'un octet qui définit le type de données qui suit l'extension : une autre extension ou un protocole de couche 4 (voir tableau Valeurs du champ "Next Header").

Pour les extensions à longueur variable, l'octet suivant contient la longueur de l'extension en mots de 8 octets, le premier n'étant pas compté (une extension de 16 octets a un champ longueur de 1).

Ordre	Charge	Next Header code
1	Basic IPv6 Header	-
2	Hop-by-Hop Options	0
3	Destination Options (with Routing Options)	60
4	Routing Header, type 0 déprécié par RFC5095	43
5	Fragment Header	44
6	Authentication Header	51
7	Encapsulation Security Payload Header	50
8	Destination Options	60
9	Mobility Header	60
-	pas de prochain en-tête	59

Les extensions peuvent s'enchaîner en suivant l'ordre défini par le [RFC 8200 Extension Header Order](#). Cette possibilité rend les politiques de filtrage plus lourdes, nécessitant une bonne connaissance du protocole et générant de la charge sur le matériel filtrant.

4. Découverte du MTU dans le chemin

La procédure "Path MTU" est définie dans le [RFC 8201](#). Son implémentation n'est pas obligatoire sur les noeuds IPv6 et peut être absente.

Initialement, l'équipement émetteur fait l'hypothèse que le PMTU (Path MTU) d'un certain chemin est égal au MTU du lien auquel il est directement attaché ($40 + 1460 = 1500$ octets).

S'il s'avère que les paquets transmis sur ce chemin excèdent la taille maximale autorisée par un lien intermédiaire, alors le routeur associé détruit ces paquets et retourne un message d'erreur ICMPv6 de type "Packet too Big", en y indiquant le MTU accepté (soit par exemple 1480). Fort de ces informations, l'équipement émetteur réduit le PMTU supposé pour ce chemin ($40 + 1440 = 1480$ octets).

Plusieurs itérations peuvent être nécessaires avant d'obtenir un PMTU permettant à tout paquet d'arriver à l'équipement destinataire sans jamais excéder le MTU de chaque lien traversé. Le protocole IPv6 garantit que le MTU de tout lien ne peut descendre en dessous de 1280 octets, valeur qui constitue ainsi une borne inférieure pour le PMTU. Le protocole reposant sur l'hypothèse de la perte de paquets, il est laissé le soin aux couches supérieures de gérer la fiabilité de la communication en retransmettant si nécessaire.

Source : [Mécanisme de découverte du PMTU](#)

A cet égard, les pare-feu pourraient bloquer ce trafic utile car il serait considéré comme une connexion directe venant d'un Internet de moindre confiance, en dehors de tout état ou session.

Aussi, ce trafic reste vulnérable à des attaques de déni de service (DoS) provoquant des ruptures de connexion ou l'impossibilité de transférer des données en envoyant des messages "Packet Too Big" à un noeud.

Le [RFC4890](#) propose des recommandations en matière de filtrage IPv6 sur les pare-feu / firewalls.

13. Introduction aux adresses IP

Ce chapitre décrit les types d'adresses IPv4 et IPv6, Unicast, Broadcast, Multicast, Anycast, adresses privées et publiques et la nécessité NAT en IPv4.

1. Adressage IP

L'adressage IP dispose des caractéristiques suivantes :

- C'est un identifiant logique (configuration administrative)
- Dont l'unicité est nécessaire : les adresses IP doivent être assignées à une seule interface.
- Qui respecte une organisation hiérarchique (par niveau) et géographique.

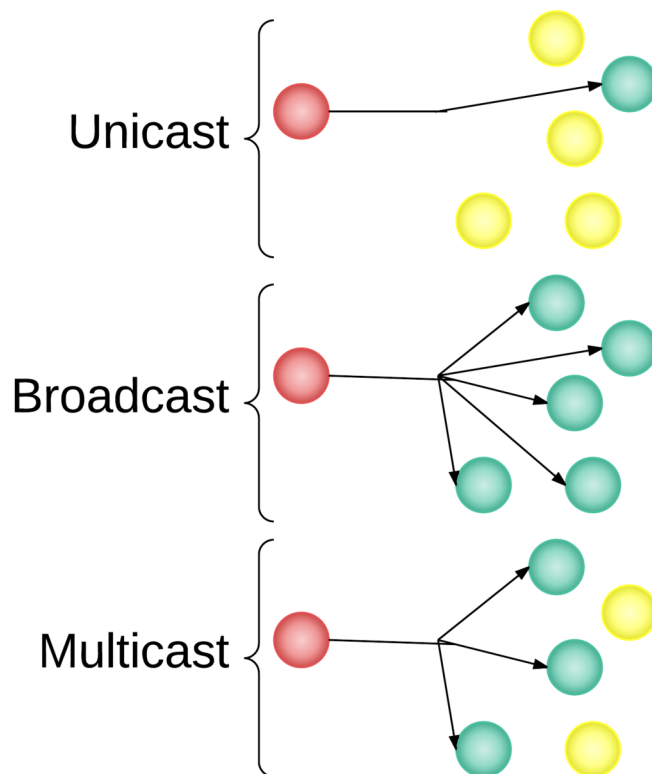
2. Type d'adresses IP

Les adresses IP permettent d'identifier de manière unique les hôtes d'origine et de destination. Les routeurs se chargent d'acheminer les paquets à travers les liaisons intermédiaires.

Il existe plusieurs types d'adresses qui correspondent à plusieurs usages.

On trouve au moins trois grandes catégories :

- les adresses *Unicast* : à destination d'un seul hôte
- les adresses *Broadcast* (IPv4) : à destination de tous les hôtes du réseau
- les adresses *Multicast* (IPv4 et IPv6) : à destination de certains hôtes du réseau.



Unicast, Broadcast et Multicast.

On peut reconnaître la nature de ces destinations en fonctions des valeurs contenues dans les adresses.

Parmi les adresses **Unicast** on distinguera :

- les adresses *non-routées* : **locales** ou de **loopback** jamais transférées par les routeurs.
- les adresses *publiques* ou *globales* : pour lesquelles les routeurs publics acheminent les paquets
- les adresses *privées* ou *unique locale* : pour lesquelles seuls les routeurs privés transfèrent le trafic (les routeurs publics ne connaissent pas de chemin pour des destinations privées).

Les adresses dites **Anycast** ne distinguent pas à vue des adresses *Unicast*. Elles désignent la destination la plus proche.

3. Internet : adressage IPv4 et IPv6

Les interfaces prennent une adresse IP :

- IPv4 : les adresses sont codées sur 32 bits (4 octets) en notation décimale pointée. Par exemple : 195 . 238 . 2 . 21. La solution IPv4 est largement épuisée aujourd'hui
- IPv6 : les adresses sont codées sur 128 bits (8 mots de 16 bits) et notées en hexadécimal. Par exemple : 200a : 14d6 : 6f8 : 1 : 7256 : 81ff : febf : 7c37

4. Masque d'adresse

En IPv4 comme en IPv6, une adresse IP est toujours accompagnée de son masque.

Le masque d'une adresse IP détermine l'appartenance d'une adresse à un réseau IP.

Un masque est une suite homogène de bits à 1 et puis de bits à 0.

On peut l'écrire :

- en notation décimale (en IPv4, ancienne méthode) : 255 . 255 . 255 . 0, par exemple
- en notation CIDR qui reprend le nombre de bits à 1 dans le masque (en IPv4 et en IPv6). Par exemple /24 = 24 bits à 1 dans le masque, soit 255 . 255 . 255 . 0.

5. Partie réseau ou préfixe

Par rapport à une adresse de référence, les bits à 1 dans le masque correspondent à la partie que toutes les adresses IP partagent si elles sont dans le même réseau.

Soit l'adresse assignée 192.168.100.100/24 :

```
192.168.100.100
255.255.255.0
```

Toutes les adresses commençant par "192.168.100.*" appartiennent au même réseau, soit 192.168.100.1/24, 192.168.100.2/24, etc. jusqu'à 192.168.100.255.

6. Partie hôte ou identifiant d'interface

Par rapport à une adresse de référence, les bits à 0 dans le masque correspondent à la partie variable d'un réseau qui identifie les hôtes de manière unique. Soit 192.168.100.100/24 :

```
192.168.100.100
255.255.255.0
```

Le nombre de bits à zéro dans le masque indique aussi le nombre d'adresses IP dans un réseau IP. Ici il y a 8 bits à zéro dans le masque, soit 2^8 (256) possibilités.

7. Première et dernière adresse IPv4

En IPv4, la première adresse est réservée à l'identification du réseau (numéro de réseau).

Également, la dernière adresse est réservée au mode de transmission *Broadcast* (toutes adresses du réseau).

Les adresses incluses entre la première et la dernière peuvent être utilisées sur les interfaces.

8. Adressage IPv6

Le principe du masque IPv4 reste valable en IPv6. Mieux, par défaut, on utilisera un masque /64 qui divise l'adresse en deux parties égales.

- La longueur de 128 bits des adresses IPv6 exige une nouvelle notation en hexadécimal.
- La première adresse est assignable car on utilise une autre manière d'identifier le réseau.
- soit l'adresse 200a : 14d6:6f8:1:7256:81ff : febf : 7c37/64 fait partie du réseau 200a : 14d6:6f8:1::/64
- Le Broadcast a disparu d'IPv6 comme mode de livraison et est remplacé par du Multicast (Well-Know ou Solicited-Node).

9. Tables et protocoles

Sur chaque hôte, on trouvera différentes tables relevant de processus ou protocoles distincts :

- Une table ARP (IPv4) et de voisinage (IPv6/ICMPv6/ND)
- Une de table de routage
- Une table de groupes Multicast joints

10. Passerelle par défaut

La passerelle par défaut est l'adresse IP dans le même réseau que l'hôte et qui permet de joindre d'autres réseaux, soit l'adresse IP du routeur dans le LAN.

- En IPv4, il faut la configurer manuellement ou l'attribuer par DHCP.
- En IPv6, elle est automatiquement annoncée par le routeur via des Router Advertisements (ND)

Elle est nécessaire car ne pouvant connaître l'adresse physique du destinataire situé dans un autre réseau, le trafic est physiquement livré à la passerelle qui décidera du sort à réserver aux paquets.

Ce paramètre devient une entrée dans la table de routage de l'hôte qui désigne le point de livraison physique pour toute destination (autre que le réseau local). Le trafic vers l'Internet sera donc livré physiquement à cette passerelle qui elle-même prendra une décision de routage.

11. Adressage privé et adressage public en IPv4

À cause du manque d'adresse IPv4 la plupart des LANs sont adressés de manière privée dans les blocs : 10.0.0.0/8, 172.16.0.0/12 et 192.168.0.0/16.

Ces adresses IPv4 privées conformément à leur nature n'ont pas de destinations sur l'Internet. Le [RFC 1918](#) définit cet espace.

Pour interconnecter un réseau privé à l'Internet public, on utilise un routeur NAT qui réalisera la traduction d'adresses privées en une ou plusieurs adresses publiques.

Expliqué de manière simple, les routeurs NAT altèrent les en-têtes en remplaçant les champs d'adresses contenant une adresse IP privée par une adresse IP publique.

La [section 2. du RFC 1918 motive l'usage des adresses privées en IPv4](#) :

Un des défis de la communauté Internet est l'épuisement des adresses globalement uniques.

12. Nécessité du NAT en IPv4

À cause de la consommation galopante d'adresses IPv4 publiques, les autorités de l'Internet ont proposé des solutions :

1. L'adoption d'un **nouveau protocole avec IPv6** corrigeant des faiblesses d'IPv4 avec un conseil de transition en double pile IPv4/IPv6.
2. En vue d'offrir une connectivité globale (publique) à des hôtes privés, une des solutions est la traduction du trafic (NAT). Une autre est la mise en place d'un **proxy applicatif**.

Ces dernières solutions de traduction tronquent le trafic IP d'origine, génèrent de la charge sur les ressources nécessaires à la traduction ou à la réécriture du trafic dans un sens et dans un autre. Ce n'est pas sans poser de problème et cela génère un coût non négligeable.

13. NAT et pare-feu

Les pare-feu (*Firewall*) ont pour objectif de filtrer les communications TCP/IP. Ils sont capables de tenir compte des sessions établies à partir d'une zone de confiance et d'empêcher tout trafic initié de l'Internet.

Quand ils sont placés dans le réseau (ailleurs que sur les hôtes), ils remplissent des tâches de routage. La plupart du temps cette fonction "pare-feu" est intégrée aux routeurs.

On a tendance à confondre les fonctionnalités NAT avec celles du pare-feu.

Même s'il est probable que le même logiciel prenne en charge les deux fonctions, ces procédures sont distinctes. **Alors que le NAT sert à traduire le trafic, il ne protège en rien.**

Cette fonction de **filtrage** est prise en charge par le pare-feu à état qui arrête les connexions non sollicitées sur le réseau à protéger. Quant à lui, le *proxy* offrira d'autres fonctions que la traduction telle que le contrôle du trafic, des mécanismes de cache, avec ou sans authentification, ...

Dans tous les cas, c'est la fonction pare-feu qui protège des tentatives de connexions externes.

14. Gestion des adresses IP

En IPv4, les adresses pourront être configurées de manière statique sur les interfaces ou dynamiquement via [DHCP \(RFC 2131\)](#).

En l'absence de service DHCP sur le réseau, les hôtes Windows et Mac OS X autoconfigurent une adresse *APIPA* (*Automatic Private Internet Protocol Addressing*) ou *IPv4LL* dans la plage d'adresses IP 169.254.0.0/16 (255.255.0.0). Ces adresses sont formalisées dans le [RFC 3927](#) sous le titre "*Dynamic Configuration of IPv4 Link-Local Addresses*"

En IPv6, on trouvera une nette amélioration des solutions de gestion des adresses avec des mécanismes d'autoconfiguration sans état (SLAAC/ND), des mécanismes de configuration dynamique, des solutions de re-numérotation des préfixes et des identifiants d'interface, des paramètres DNS (DHCPv6, RDNSS). Toutefois, comme en IPv4, ces procédures et protocoles restent vulnérables à des attaques locales.

14. Adressage IPv4

Il est inimaginable de se présenter à un examen Cisco ou à un entretien d'embauche dans le domaine des réseaux sans maîtriser l'adressage IPv4 et ses mécanismes de découpage. On trouvera ici un exposé sur ce sujet.

1. Introduction aux adresses IPv4

Une adresse IPv4 est un identifiant de 32 bits représentés par 4 octets (8 bits) codés en décimales séparées par des points.

Le masque de réseau lui aussi noté en décimal pointé indique avec les bits à 1 la partie réseau partagée par toutes les adresses d'un bloc et avec les bits à 0 la partie unique qui identifie les interfaces sur la liaison.

Par exemple, 192.168.1.255.255.255.0 indique un numéro de réseau (première adresse) 192.168.1.0 et un numéro de Broadcast 192.168.1.255. Toutes les adresses comprises entre ces valeurs peuvent être utilisées par les interfaces attachées à une même liaison (un même switch).

À cause du manque d'espace IPv4 disponible, on trouve souvent des masques qui chevauchent les octets, ce qui nécessite de passer par des calculs binaires.

2. Définition

- Une adresse IPv4 (Internet Protocol version 4) est une identification unique pour un hôte sur un réseau IP.
- Une adresse IP est un nombre d'une valeur de 32 bits représentée par 4 valeurs décimales pointées ; chacune a un poids de 8 bits (1 octet) prenant des valeurs décimales de 0 à 255 séparées par des points. La notation est aussi connue sous le nom de "décimale pointée".

3. Identification de la classe d'adresse

(RFC791)

Les Classes

À l'origine d'IPv4, on distingue une organisation en classes d'adresses dont les quatre premiers bits indiquent la classe.

- Les adresses de Classe A commencent par 0xxx en binaire, ou 0 à 127 en décimal.
- Les adresses de Classe B commencent par 10xx en binaire, ou 128 à 191 en décimal.
- Les adresses de Classe C commencent par 110x en binaire, ou 192 à 223 en décimal.
- Les adresses de Classe D commencent par 1110 en binaire, ou 224 à 239 en décimal.
- Les adresses de Classe E commencent par 1111 en binaire, ou 240 à 255 en décimal.

Notes sur les Classes d'adresses :

- Seules les adresses de Classes A, B et C sont assignables à des interfaces (**adresse d'Unicast**)

- La classe D est utilisée pour des **adresses de Multicast** (adresse unique identifiant de nombreuses destinations)
- La classe E est utilisée pour des besoins futurs ou des objectifs scientifiques

Adresses spécifiques :

- Les adresses commençant de 127.0.0.0 à 127.255.255.255 sont réservées pour le bouclage (loopback)
- Adresses privées non routables vers l'Internet sont ([RFC1918](#)) :
 - Pour la classe A : de 10.0.0.0 à 10.255.255.255
 - Pour la classe B : de 172.16.0.0 à 172.31.255.255
 - Pour la classe C : de 192.168.0.0 à 192.168.255.255

Distinction de la partie réseau de la partie hôte

Par défaut :

- La partie réseau des **adresses de Classe A** portera sur le premier octet et la partie hôte sur les trois derniers ($2^{24} = 16\,777\,216$ hôtes possibles par réseau)
- La partie réseau des **adresses de Classe B** portera sur les deux premiers octets et la partie hôte sur les deux derniers ($2^{16} = 65\,536$ hôtes possibles par réseau)
- La partie réseau des **adresses de Classe C** portera sur les trois premiers octets et la partie hôte sur le dernier ($2^8 = 256$ hôtes possibles par réseau)

Partie Réseau

Partie Hôte

Adresse de la Classe A	100 . 150 . 25 . 3	2 exp 24 = 16 777 216 hôtes possibles par sous-réseaux
Adresse de la Classe B	136 . 10 . 100 . 25	2 exp 16 = 65 536 hôtes possibles par sous-réseaux
Adresse de la Classe C	195 . 74 . 212 . 12	2 exp 8 = 256 hôtes possibles par sous-réseaux

Exemple d'adresses IP avec les hôtes possibles dans ce réseau, par défaut

Classes d'adresses IPv4

4. Utilisation d'un masque

Utilisation d'un masque ([RFC950](#))

Un masque va préciser de manière certaine dans quel réseau se trouve une adresse IP et en conséquence :

1. L'**adresse du réseau** (appelée aussi numéro de réseau, non assignable)
2. L'**adresse de Broadcast** (adresse visant toutes les destinations, non assignable)
3. La plage d'adresses utilisables (de la première à la dernière en dehors des adresses précitées)

Un masque sera une suite de 32 bits divisée en 4 octets pointés composée uniquement d'abord d'une suite de 1 et, après, d'une suite de 0. La notation est aussi décimale pointée. Toutefois, on trouvera une autre notation dite CIDR (*Classless Interdomain Routing*) qui représente le nombre de bits pris par la partie réseau du masque.

Masque par défaut

Le nombre d'hôtes possibles obtenus ci-dessus correspond à l'application d'un masque par défaut sur un type de classe d'adresse :

- Le masque par défaut des adresses de Classe A est 255.0.0.0 ou /8
- Le masque par défaut des adresses de Classe B est 255.255.0.0 ou /16
- Le masque par défaut des adresses de Classe C est 255.255.255.0 ou /24

5. Méthode par calcul binaire

L'adresse du réseau, l'adresse de Broadcast et la plage d'adresses utilisables peuvent être obtenues à partir d'un calcul booléen de type ET ou la conjonction logique (une proposition est vraie lorsque les deux termes sont tous les deux vrais) :

a. Obtenir l'adresse du réseau :

Pour l'adresse IP 140.159.125.25, adresse de classe B à laquelle on applique un masque par défaut de 255.255.0.0 :

```
10001100.10011111.01111101.00011001 140.159.125.25
11111111.11111111.00000000.00000000 255.255.0.0
-----
10001100.10011111.00000000.00000000 140.159.0.0
```

L'adresse du réseau est donc 140.159.0.0. Elle est la première adresse de la plage.

b. Obtenir l'adresse de Broadcast :

On va remplacer les bits de valeur 0 de la partie hôte du résultat obtenu pour l'adresse de réseau par des bits de valeur 1, soit les deux derniers octets maximisés :

```
10001100.10011111.00000000.00000000 140.159.0.0
^^^^
```

par :

```
10001100.10011111.11111111.11111111 140.159.255.255 ““
```

c. Obtenir la plage d'adresses de ce réseau :

La plage d'adresse du réseau sera comprise entre la première adresse utilisable et la dernière utilisable, autrement dit, celle qui suit l'adresse du réseau et celle qui précède l'adresse de Broadcast :

De

```
10001100.10011111.00000000.00000001 140.159.0.1
```

à

```
10001100.10011111.11111111.11111110 140.159.255.254
```

Masques restrictifs

Les masques présentés ci-dessus sont des masques appartenant par défaut à chaque classe d'adresse. On pourra utiliser d'autres masques plus restrictifs afin de diviser un réseau en plusieurs sous-réseaux afin d'optimiser un plan d'adressage.

On va emprunter des bits à la partie hôte au profit de la partie réseau. De manière intuitive, on peut considérer qu'à partir d'un réseau de classe C de 256 adresses possibles, on pourra, par exemple obtenir 4 sous-réseaux différents de 64 adresses. Dans ce cas-ci, on dira que l'on a emprunté deux bits à la partie hôtes ($2 \exp 2 = 4$) au profit des sous-réseaux. Il ne reste que 6 bits pour les hôtes ($2 \exp 6 = 64$) dans chacun de ces sous-réseaux.

Par exemple :

Pour l'adresse IP 195.74.212.78, adresse de classe C à laquelle on applique un masque de 255.255.255.192 par la même méthode que présenté ci-avant :

a. Obtenir l'adresse du réseau :

```
11000011.01001010.11010100.01001110 195.74.212.78
11111111.11111111.11111111.11000000 255.255.255.192
-----
11000011.01001010.11010100.01000000 195.74.212.64
```

L'adresse du réseau est donc 195.74.212.64

b. Obtenir l'adresse de Broadcast :

On va remplacer les bits de valeur 0 de la partie hôte du résultat obtenu pour l'adresse de réseau par des bits de valeur 1 :

```
11000011.01001010.11010100.01000000 195.74.212.64
```

par :

```
11000011.01001010.11010100.01111111 195.74.212.127
```

c. Obtenir la plage d'adresses de ce réseau :

La plage d'adresse du réseau sera comprise entre la première adresse utilisable et la dernière utilisable, autrement dit, celle qui suit l'adresse du réseau et celle qui précède l'adresse de Broadcast :

De

```
11000011.01001010.11010100.01000001 195.74.212.65
```

à

```
11000011.01001010.11010100.01111110 195.74.212.126
```

6. Méthode dite du nombre magique.

Le calcul binaire peut sembler fastidieux. La méthode dite du nombre magique permet d'éviter ces calculs.

Le nombre magique est 256 soustrait de la valeur intéressante autre que 0 ou 255 du masque.

Pour trouver l'adresse réseau, il suffira de **trouver le multiple du nombre magique directement inférieur ou égal à l'adresse IP**, un multiple étant le résultat d'une multiplication :

Pour l'adresse IPv4 195.74.212.136, adresse de classe C à laquelle on applique un masque de 255.255.255.192, le nombre magique est $256-192 = 64$, le multiple juste inférieur étant 128. L'adresse réseau est donc 195.74.212.128.

Pour trouver l'adresse de la première adresse utilisable, il faudra ajouter 1 au dernier octet du numéro de sous-réseau : 195.74.212.129.

Pour trouver l'adresse de Broadcast, il faudra faire (numéro de sous-réseau + nombre magique - 1) $128+64-1$, ce qui donnera l'adresse 195.74.212.191.

Pour trouver l'adresse de la dernière adresse utilisable, il faudra soustraire 1 au dernier octet de l'adresse de Broadcast : 195.74.212.190.

Autre exemple plus complexe :

	Octet 1	Octet 2	Octet 3	Octet 4	Commentaire
Adresse IPv4	10	200	10	18	
Masque	255	224	0	0	
Numéro de sous-réseau	10	192	0	0	Nombre magique = $256-224 = 32$
Première adresse utilisable	10	192	0	1	Ajouter 1 au dernier octet du numéro de sous-réseau
Adresse de Broadcast	10	223	255	255	$192+32-1 = 223$
Dernière adresse utilisable	10	223	255	254	Soustraire 1 du dernier octet de l'adresse de Broadcast

Nombre de sous-réseaux / nombre d'hôtes

Les formules sont simples :

- Nombre de sous-réseaux : $2^{\text{EXP bits empruntés pour les sous-réseaux}} - 2$
- Nombre d'hôtes : $2^{\text{EXP bits restant pour les hôtes}} - 2$

À condition d'avoir en tête ce tableau de conversion, ce qui est indispensable :

|Bits | Binaire | Décimal — | — | — 0 | 00000000 | 0 1 | 10000000 | 128 2 | 11000000 | 192 3 | 11100000 | 224 4 | 11110000 | 240 5 | 11111000 | 248 6 | 11111100 | 252 7 | 11111110 | 254 8 | 11111111 | 255

7. Le routage sans classe (CIDR)

Les [RFC1518](#) (historique) et [RFC4632](#) (technique) consacrent la méthode CIDR, routage sans classe interdomaine, comme méthode de gestion et d'organisation plus efficace des adresses Internet. Concrètement cela signifie que :

- Une adresse est toujours accompagnée de son masque identifiant son appartenance à un réseau (domaine de Broadcast)
- La notion de classe d'adresse IPv4 disparaît dans la pratique.
- La notion de blocs identifiant des domaines de routage remplace la notion de classe d'adresse IPv4, rendant l'usage d'un masque indispensable.
- Pour simplifier la notation, on préfère représenter un slash / + le nombre de bits à 1 dans le masque. Par exemple /26 au lieu de 255.255.255.192.
- Les blocs sont variables, peuvent être agrégés ou découpés dans les informations de routage ou dans les plans d'adressage.
- Ces blocs sont des ensembles homogènes (les adresses se suivent) d'adresses en base 2 : 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, ..., 65536, ... adresses.

- Le [RFC950](#) reste une référence en matière de calculs (numéro de réseau, Broadcast, plage d'adresses).
- Les protocoles de routage doivent supporter le masque comme information supplémentaire. On parle de protocoles de routage *classless* rendant RIPv1 et IGRP obsolètes.
- Toute une série de questions peuvent être posées dans la maîtrise de l'adressage IPv4 et peuvent être résolues soi-même ou avec un logiciel pour automatiser des tâches.

8. Notation CIDR

CIDR : Classless Inter-Domain Routing

Au lieu de représenter le masque d'une adresse en notation décimale pointée, le CIDR propose de noter une adresse suivie de "/" nombre de bits à 1, le nombre de bits à zéro qui reste représente la taille du bloc. Soit, par exemple :

255.255.255.192 $\Rightarrow$ 26 bits à 1 $\Rightarrow$ /26

Il est utile de connaître les masques par défaut :

- 255.0.0.0 : /8
- 255.255.0.0 : /16
- 255.255.255.0 : /24

Le masque donne aussi le nombre de bits à zéro (32 - le masque).

- Par exemple, un /20 fournit un bloc de 2^{12} adresses = 4096.
- Par exemple, un /24 fournit un bloc de 2^8 adresses = 256.
- Par exemple, un /30 fournit un bloc de 2^2 adresses = 4. Les bits à zéro dans le masque indiquent l'étendue du réseau.

9. Masques à longueurs variables (VLSM)

([RFC1878](#))

Dans l'ancienne méthode, les masques de découpage devaient être identiques parce qu'ils n'étaient pas transportés dans les informations de routage.

Rappels

Il est utile de rappeler que dans un masque, les bits à 1 correspondent à la partie fixe qui identifie le réseau et les bits à zéro correspondent à la partie variable qui identifie précisément une interface (un hôte, un noeud) dans ce domaine.

La première adresse est réservée et identifie le réseau en général. Elle n'est pas utilisable sur une interface mais identifie les réseaux dans les tables de routage.

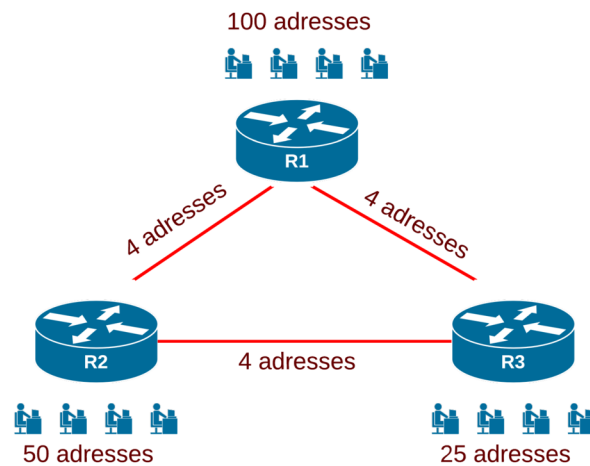
La dernière ne peut pas se configurer sur une interface et est utilisée comme adresse de destination pour la diffusion (Broadcast).

Pour bien comprendre le mécanisme, il est plus intéressant de se représenter une topologie censée être fixée. Dans la réalité, il faudra tenir compte de l'évolutivité des besoins en adresses étant donné qu'un réseau que l'on planifie ne décroît jamais par définition. Le quotidien de l'administration des réseaux consiste en la gestion des adressages privés (cachés par du NAT). Pourquoi se passer d'un bloc 192.168.0.0/16, 172.16.0.0/12 ou 10.0.0.0/8 que l'on découpera aisément ?

Premier cas d'école

On peut partir d'un ancien cas d'école considérant qu'il faille adresser un interrèseau avec des adresses globales (publiques). On vous octroie un bloc d'adresses IPv4 195.167.46.0/24. Selon les contraintes représentées d'un internet maillé de trois routeurs ayant chacun un réseau local (LAN) :

- LAN de R1 100 PCs,
- LAN de R2 50 PCs,
- LAN de R3 25 PCs.

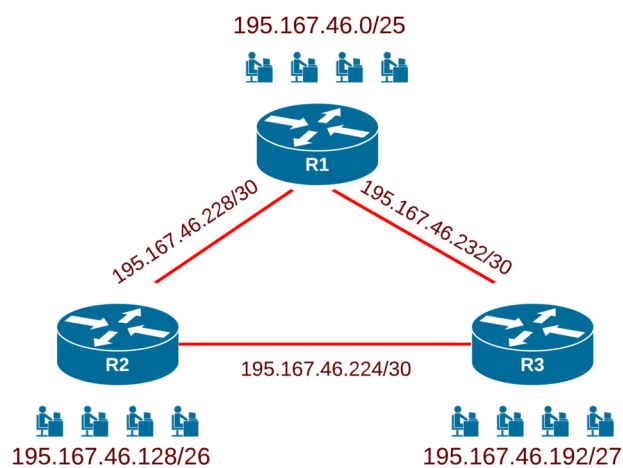


Tripod exigences

On a donc un bloc de 256 adresses (/24). On propose ici de le découper de la manière suivante :

- un bloc /25 de 128 adresses pour le LAN de R1,
- un bloc /26 de 64 adresses pour le LAN de R2,
- un bloc /27 de 32 adresses pour le LAN de R3.

Dans le reste, on prendra trois blocs /30 de 4 adresses pour adresser les connexions point à point.



Tripod découpage VLSM

- La première adresse du bloc est attribuée comme numéro de réseau au réseau LAN de R1 car il a les plus gros besoins : 195.167.46.0. Pour trouver la plage du réseau LAN de R1 (100 adresses), on fixe les bits à zéro dans le masque : 7 bits à zéro (2 EXP 7 = 128 adresses) soit de 195.167.46.0/25 à 195.167.46.127/25.
- La prochaine adresse est 195.167.46.128. Elle est le numéro de réseau du prochain réseau ayant les plus gros besoins en adresses : le LAN de R2. Le masque a besoin de 6 bits à zéro (2 EXP 6 = 64) pour le LAN de R2 (50 adresses), soit de 195.167.46.128/26 à 195.167.46.191/26
- 64 adresses plus loin, 195.167.46.192 est la première adresse (le numéro de réseau) du LAN de R3. Le LAN de R3 a besoin de 25 adresses en offrant un masque avec 5 bits à zéro (2 EXP 5 = 32 adresses) : de 195.167.46.192/27 à 195.167.46.223/27.
- 195.167.46.224 est la prochaine adresse disponible. Il reste un bloc de 32 adresses, jusqu'à 195.167.46.255. Les connexions point à point prennent un masque /30 comprenant 4 adresses dont 2 utiles : l'une pour le numéro de réseau, deux utiles, la dernière pour la diffusion.

Le plan d'adressage proposé est représenté en tableau.

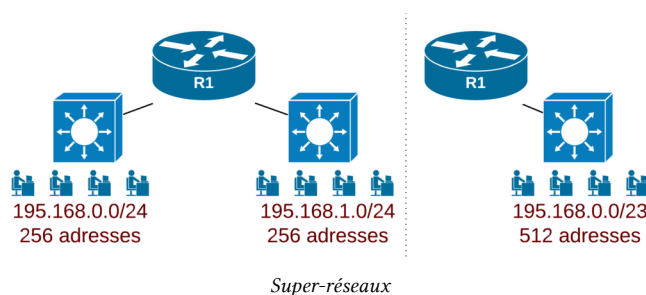
Attribution	Contrainte	Réseau	Plage adressable	Diffusion
—	—	—	—	—
LAN R1	100 adresses	195.167.46.0/25	195.167.46.1/25 à 195.167.46.126/25	195.167.46.127/25
LAN R2	50 adresses	195.167.46.128/26	195.167.46.129/26 à 195.167.46.190/26	195.167.46.191/26
LAN R3	25 adresses	195.167.46.192/27	195.167.46.193/27 à 195.167.46.222/27	195.167.46.223/27
R2-R3	4 adresses	195.167.46.224/30	195.167.46.225/30 à 195.167.46.226/30	195.167.46.227/30
R1-R2	4 adresses	195.167.46.228/30	195.167.46.229/30 à 195.167.46.230/30	195.167.46.231/30
R1-R3	4 adresses	195.167.46.232/30	195.167.46.233/30 à 195.167.46.234/30	195.167.46.235/30

Cette perspective restrictive correspond aux principes de conservation des adresses IPv4 publiques.

10. Super-réseaux

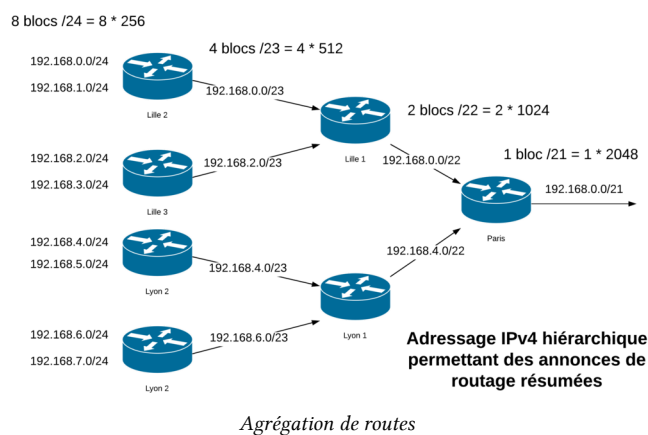
Ce qu'on appelle communément le *super-netting* consiste à regrouper des blocs contigus pour créer un seul bloc plus large.

Par exemple, pour adresser un réseau de 500 machines on peut prendre deux /24, soit 2 X 256. Le masque devient /23. En effet, /23 nous offrent 9 bits à 0, soit 2 EXP 9 = 512 adresses. Le CIDR a permis d'octroyer des blocs larges aux derniers FAI IPv4 indépendamment des classes d'adresses C.



11. Agrégation de routes

Afin de diminuer la taille des tables de routage, alors que le comportement courant est de trouver une entrée pour chaque réseau, on peut "résumer" les routes à condition d'avoir un plan d'adressage qui regroupe les réseaux géographiquement. Considérons par exemple la topologie d'un réseau privé entre Paris et six sites distants : plusieurs réseaux concentrés sur Lille et sur Lyon. Les réseaux d'extrémité contiennent deux VLANs : l'un pour les données et l'autre pour la voix.



Le routeur à Paris devrait voir deux routes dans sa table de routage l'une annonçant 192.168.0.0/22 passant par Lille et une autre annonçant 192.168.4.0/22 passant par Lyon.

12. Protocoles de routage sans classe (Classless)

Le CIDR doit être supporté par les protocoles de routage :

- RIPv2
- OSPFv2 et OSPFv3
- EIGRP
- BGPv4

13. Exercices IPv4

On peut calculer les plans d'adressage et répondre à plusieurs questions :

- Déterminer la plage IP à partir d'une adresse et de son masque
- Comparer deux adresses IP afin de déterminer si elles sont dans la même plage
- Choisir un masque adapté aux contraintes en nombre d'hôtes, avec une marge de croissance
- Choisir un masque adapté aux contraintes en nombre de réseaux, avec une marge de croissance
- Planifier un adressage avec masques fixes
- Planifier un adressage avec masques à longueurs variables
- Calculer une summarization de route

Outil pour vérifier ses calculs

La librairie Python netaddr est intéressante à cet égard : <https://netaddr.readthedocs.org/en/latest/>

L'utilitaire ipcalc calcule pour nous les adresses IP :

ipcalc --help

Usage: ipcalc [OPTION...]

```

-c, --check          Validate IP address for specified address family
-4, --ipv4           IPv4 address family (default)
-6, --ipv6           IPv6 address family
-b, --Broadcast      Display calculated Broadcast address
-h, --hostname       Show hostname determined via DNS
-m, --netmask        Display default netmask for IP (class A, B, or C)
-n, --network        Display network address
-p, --prefix         Display network prefix
-s, --silent         Don't ever display error messages

```

Help options:

```

-?, --help          Show this help message
--usage             Display brief usage message

```

Par exemple :

ipcalc -n -b -m 192.167.87.65/26

NETMASK=255.255.255.192

BROADCAST=192.167.87.127

NETWORK=192.167.87.64

On trouvera un utilitaire plus explicite avec sipcalc, il est en dépôt chez Debian/Ubuntu. Sous Centos/RHEL, il sera nécessaire de le compiler soi-même (<http://www.routemeister.net/projects/sipcalc/files/sipcalc-1.1.6.tar.gz>).

sipcalc -I ens33

-[int-ipv4 : ens33] - 0

[CIDR]

```

Host address          - 172.16.98.241
Host address (decimal) - 2886755057
Host address (hex)    - AC1062F1
Network address       - 172.16.98.0
Network mask          - 255.255.255.0
Network mask (bits)   - 24
Network mask (hex)    - FFFFFFF0
Broadcast address     - 172.16.98.255
Cisco wildcard        - 0.0.0.255
Addresses in network  - 256
Network range         - 172.16.98.0 - 172.16.98.255
Usable range          - 172.16.98.1 - 172.16.98.254

```

Liens pour s'exercer :

- [Quiz Adressage IPv4 niveau 1](#)
- [Quiz Adressage IPv4 niveau 2](#)

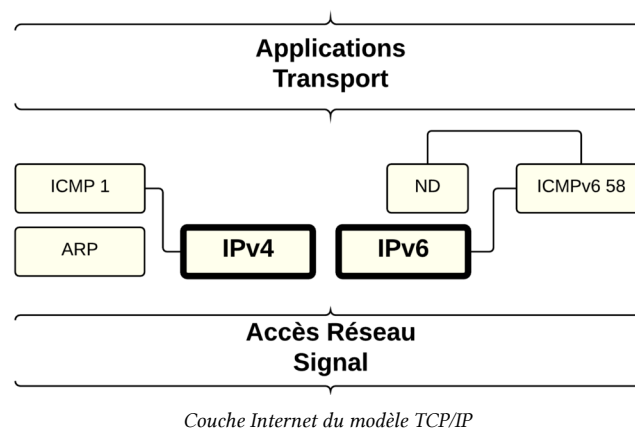
15. Protocoles ARP et ICMP

1. Introduction

1.1. ARP, ICMP, ICMPv6

IPv4 est aidé par deux protocoles pour la résolution d'adresses et le contrôle : ARP et ICMP.

En IPv6, c'est ICMPv6 qui remplit ces deux fonctions.



1.2. Protocoles de résolution d'adresses et de découverte du voisinage

- Afin d'encapsuler un paquet IP dans une trame, l'hôte d'origine a besoin de connaître l'adresse physique (MAC) de la destination.
- En IPv4, c'est le protocole ARP (Address Resolution Protocol) qui remplit cette fonction. Les hôtes IPv4 maintiennent une table de correspondance entre les adresses IPv4 à joindre et leur adresse physique (MAC). Cette table est appelée "cache ARP".
- En IPv6, c'est le protocole ND (Neighbor Discovery), sous-protocole ICMPv6, qui reprend cette fonction. Les hôtes IPv6 maintiennent une table de correspondance entre les adresses IPv6 à joindre et leur adresse physique. Elle est appelée "table de voisinage".

Commandes utiles

Table ARP sous Windows et Linux * `arp -a`

Table de voisinage sous Linux

- `ip -6 neigh`

Table de voisinage sous Windows

- `netsh interface ipv6 show neighbors`

2. Résolution d'adresse ARP

La résolution d'adresse est utile dans les réseaux IPv4 pour obtenir l'adresse physique à laquelle une adresse IP correspond.

On peut obtenir le cache ARP sur un système avec la commande :

```
arp -a
```

Ce processus permet à l'hôte émetteur de trouver l'adresse de livraison physique du trafic.

Il permet à l'hôte d'encapsuler le trafic au niveau de la couche Accès Réseau (Liaison de données) en ajoutant l'adresse MAC du destinataire dans l'en-tête de la trame.

2.1. Address Resolution Protocol

ARP est le protocole de résolution d'adresse utilisé par IPv4.

- Il est directement encapsulé par la couche 2.
- Il est donc indépendant d'IP.
- Il est formalisé par le [RFC 826](#).

2.2. Résolution d'adresses en IPv4

Au moment de l'encapsulation d'un paquet IPv4 dans une trame Ethernet ou Wi-Fi par exemple, l'hôte émetteur connaît d'avance l'adresse IP de destination. Mais comment peut-il connaître son adresse physique correspondante (l'adresse MAC de destination par exemple) afin de placer le trafic sur le support ?

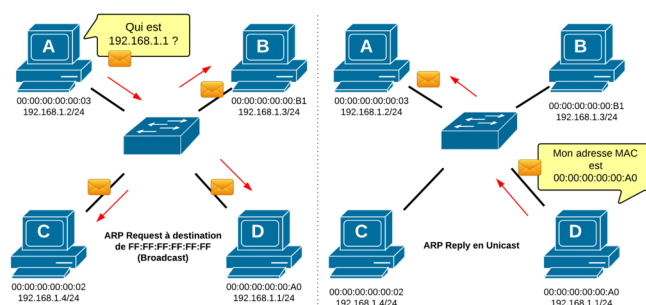
Un hôte TCP/IP ne peut connaître l'adresse de destination sans qu'elle ne s'annonce elle-même de *manière gratuite* ou de *manière sollicitée*.

Dans le but de maintenir une correspondance entre des adresses IP à joindre et leur adresse physique de destination, les hôtes TCP/IP entretiennent une **"table ARP"** pour les adresses IPv4 et une **"table de voisinage"** pour les adresses IPv6.

ARP est un protocole indépendant d'IPv4 qui offre ce service de résolution d'adresses.

En IPv6, ce sont des paquets ICMPv6 appelés Neighbor Discovery (ND) qui sont utilisés selon un mode sensiblement différent. En IPv6, les fonctions d'informations et de contrôle (ICMPv6) ont été améliorées et renforcées.

La requête ARP émane en Broadcast et la réponse est envoyée en unicast. ND (IPv6) aura un fonctionnement similaire en utilisant une adresse Multicast spéciale en lieu et place du Broadcast.



Résolution d'adresse par ARP

On trouvera une capture de trafic de ce processus de résolution d'adresse par ARP : <http://www.cloudshark.org/captures/96a2bb5fe747?>

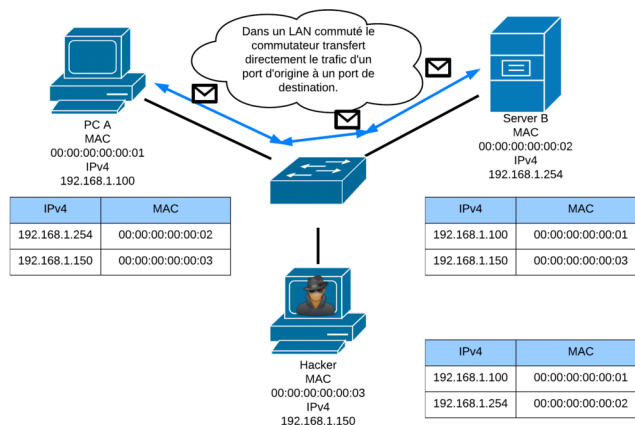
Ce trafic émane en Broadcast et se termine en Unicast.

2.3. Variantes ARP

- ARP Probe
- Gratuitous ARP : annonces sans état

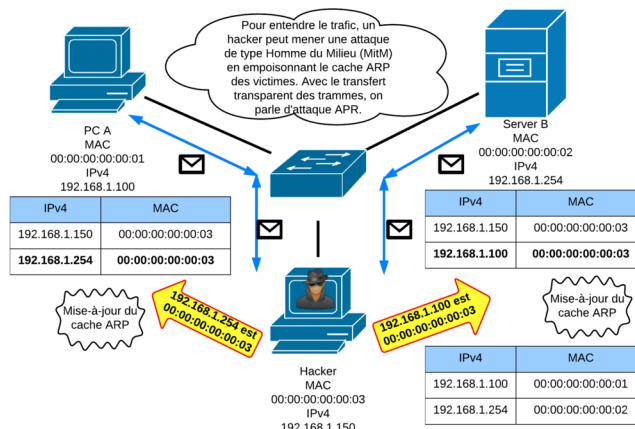
- Inverse ARP : obtenir l'IP à partir l'adresse L2 (Frame-Relay)
- Reverse ARP : attribution d'adresse IP
- Proxy ARP : mandataire ARP (routeur), fonction que l'on conseille de désactiver.

2.4. Processus ARP



Processus ARP dans un environnement commuté

2.5. Empoisonnement de cache ARP



Attaque ARP Poison Routing (APR)

3. ICMP

En IPv4, ICMP "aide" IP par des messages de contrôle et d'erreur. ICMP est formalisé dans le [RFC792](#).

Communément, il y a deux classes de messages ICMP :

- Les messages d'erreur dont le type est de 0 à 127.
- Les messages d'information dont le type est de 128 à 255.

Les différents types ICMP sont précisés par des codes.

3.1. Liste de messages ICMP

Voici une liste non exhaustive des types de messages ICMP :

Type	Message
Type 0	réponse d'écho
Type 1 et 2	réservés
Type 3	destinataire inaccessible
Type 4	extinction de la source
Type 5	redirection
Type 8	écho
Type 11	temps dépassé
Type 12	en-tête erroné
Type 13	demande heure
Type 14	réponse heure
Type 15	demande adresse IP
Type 16	réponse adresse IP
Type 17	demande masque sous-réseau
Type 18	réponse masque sous-réseau

3.2. ICMP Types 8 Echo Request - 0 Echo Reply

Le programme ping qui permet entre autre de vérifier la connectivité, envoie un paquet "ICMP Echo request" (type 8 code 0), peut recevoir un paquet "ICMP Echo Reply" (type 0 code 0).

Capture de paquets ICMP à la suite de la commande >ping www.google.com : <https://www.cloudshark.org/captures/f1b5fba6f83f>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.10	172.217.17.68	ICMP	98	Echo (ping) request id=0x4ade, seq=0/0, ttl=64
2	0.013427	172.217.17.68	192.168.0.10	ICMP	98	Echo (ping) reply id=0x4ade, seq=0/0, ttl=53 (request in 1)

```

↳ Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
↳ Ethernet II, Src: Apple_95:7f:5b (ac:bc:32:95:7f:5b), Dst: Netgear_lb:65:6a (20:e5:2a:1b:65:6a)
↳ Internet Protocol Version 4, Src: 192.168.0.10, Dst: 172.217.17.68
↳ Internet Control Message Protocol
  Type 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xaa62 [correct]
    Identifier (BE): 19166 (0x4ade)
    Identifier (LE): 56906 (0xde4a)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
    Timestamp from icmp data: Oct 9, 2016 16:21:19.938292000 UTC
    Timestamp from icmp data (relative): 0.000051000 seconds
  ↳ data

```

Message ICMP type 8

3.3. ICMP Type 3 Destination Unreachable

Les messages "ICMP Type 3 Destination Unreachable" ont des codes qui ne sont pas toujours rendus par les logiciels.

Code	Signification
0	Destination network unreachable
1	Destination host unreachable
2	Destination protocol unreachable
3	Destination port unreachable
4	Fragmentation required, and DF flag set
5	Source route failed
6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Network administratively prohibited
10	Host administratively prohibited
11	Network unreachable for ToS
12	Host unreachable for ToS
13	Communication administratively prohibited
14	Host Precedence Violation
15	Precedence cutoff in effect

3.4. Type 11 Time exceeded

A la suite de la commande `tracert -m 1 www.google.com`, on un message “ICMP Type 11 Time exceeded” revenant du premier saut :

Capture : <https://www.cloudshark.org/captures/f751ade6b3f1>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.1	192.168.0.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2	0.002143	192.168.0.1	192.168.0.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
3	0.003551	192.168.0.1	192.168.0.10	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)

```

↳ Frame 1: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
↳ Ethernet II, Src: CiscoInc_3d:24:19 (00:ca:e5:3d:24:19), Dst: Apple_95:7f:5b (ac:bc:32:95:7f:5b)
↳ Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.10
↳ Internet Control Message Protocol
  Type: 11 (Time-to-live exceeded)
  Code: 0 (Time to live exceeded in transit)
  Checksum: 0x7401 [correct]
  ↳ Internet Protocol Version 4, Src: 192.168.0.10, Dst: 172.217.17.68
  ↳ User Datagram Protocol, Src Port: 56948 (56948), Dst Port: 33435 (33435)

```

Message ICMP type 11

3.5. Conclusions

- Quand du trafic TCP/IP doit être envoyé, l’hôte encapsule le trafic à destination de l’adresse MAC apprise par ARP.
- Les messages “ARP Reply” peuvent être gratuits.
- ICMP complète IPv4 en terme de contrôle et de messages d’erreurs.

16. Couche Transport TCP et UDP

Les protocoles TCP et UDP de la couche Transport des modèles OSI et TCP/IP, vus de manière comparative, font partie des sujets vérifiés dans les certifications ICND1 et CCNA. On trouvera dans ce chapitre une tentative d'éclairage sur le sujet.

1. Introduction

Les protocoles de la couche de transport peuvent résoudre des problèmes comme la fiabilité des échanges ("est-ce que les données sont arrivées à destination?") et assurer que les données arrivent dans l'ordre correct.

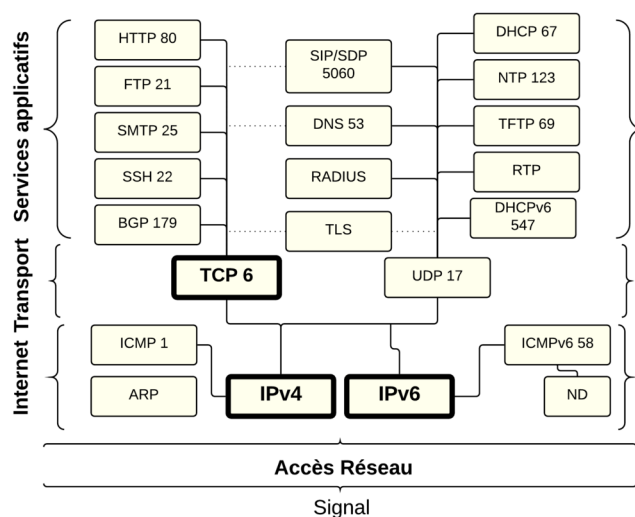
Dans la suite de protocoles TCP/IP, les protocoles de transport déterminent aussi à quelle application chaque paquet de données doit être délivré.

Les protocoles de couche transport font le lien entre les hôtes (IP) et les services applicatifs (HTTP, FTP, DNS, et d'autres).

Mais on retiendra principalement que la couche transport est responsable des dialogues (sessions) entre les hôtes terminaux. Elle permet de multiplexer les communications entre les noeud IP en offrant un support à la couche application de manière :

- Fiable et connectée avec le protocole TCP
- Non-fiable et non-connectée avec le protocole UDP

Les ports TCP ou UDP (65536 sur chaque interface) permettent aux hôtes terminaux d'identifier les dialogues.



Couches Internet, Transport et Application du modèle TCP/IP

1.1. TCP

TCP, *Transmission Control Protocol*, offre des services d'établissement et de fin de dialogue ainsi que des messages de maintenance de la communication en mode fiable et connecté avec :

- des accusés de réception

- du séquençage, de l'ordonnancement
- du contrôle de flux (fenêtrage)
- de la reprise sur erreur
- du contrôle de congestion
- de la temporisation

1.2. UDP

UDP, *User Datagram Protocol*, s'occupe uniquement du transport non fiable.

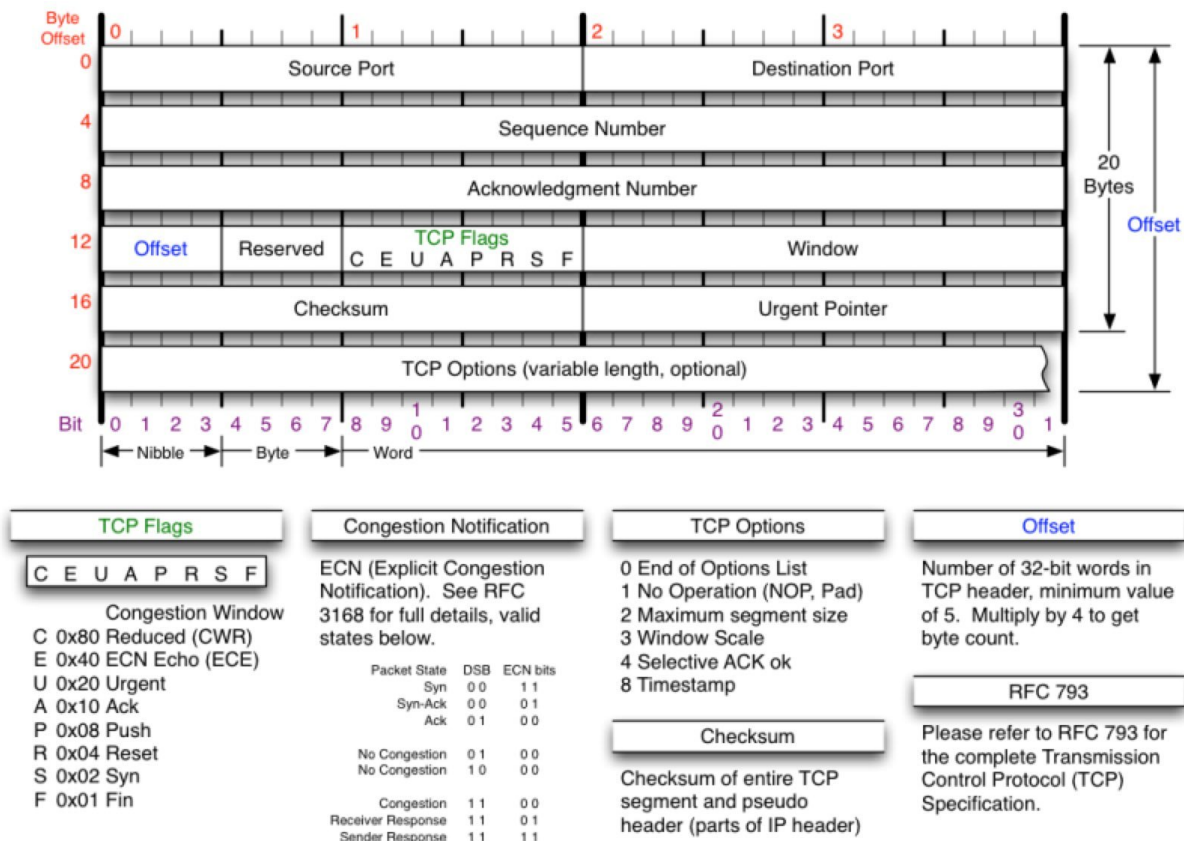
- Il est une simple passerelle entre IP et l'application.
- Il est conseillé pour les applications pour du trafic en temps réel à taille fixe et régulier (voix, vidéo).
- Il supporte des protocoles simples (TFTP, SNMP) ou souffrant des délais (DHCP, DNS, NTP).

1.3. Comparaison UDP et TCP

Il est utile de comparer UDP et TCP :

- UDP est un en-tête amoindri des fonctionnalités TCP.
- UDP dispose presque uniquement des champs ports source et port destination.

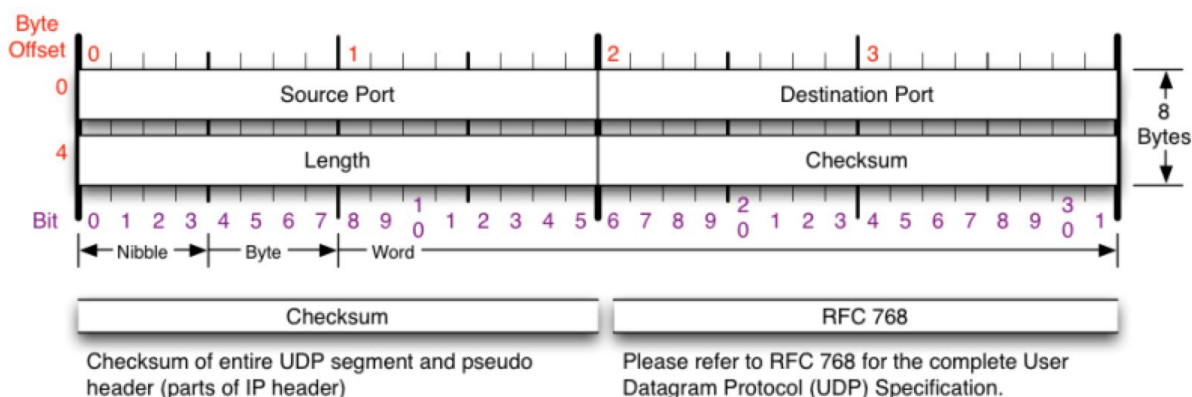
1.4. En-tête TCP



En-tête TCP

Source de l'image : TCP/IP Reference nmap.org

1.5. En-tête UDP



En-tête UDP

Source de l'image : TCP/IP Reference nmap.org

2. Numéros de ports

Les ports sont des portes d'entrée entre les hôtes terminaux. Ils sont codés sur 16 bits de 0 à 65535.

Un client IP ouvre un port à l'origine à destination d'un serveur IP écoutant sur un port de destination. La réponse émane du port de l'application sur le serveur à destination du couple IP :port client ouvert à l'origine.

La commande `netstat -a` sur un PC permet de connaître tous les ports à l'écoute et les liaisons maintenues à l'instant (UDP et TCP sur IPv4 et IPv6).

C'est ce qu'on appelle un "socket" : l'adresse IP combinée au port identifie chaque partenaire de communication.

La liste complète des ports se trouve ici : [Service Name and Transport Protocol Port Number Registry](#).

Port TCP/UDP par défaut	Protocole
TCP 20	FTP transfert de données
TCP 21	FTP signalisation
TCP 22	SSH
TCP 23	Telnet
TCP 25	SMTP
UDP/TCP 53	DNS
UDP 67	BOOTPs (DHCP server)
UDP 68	BOOTPc (DHCP client)
UDP 69	TFTP
TCP 80	HTTP
TCP 110	POP3
UDP 123	NTP
UDP 161	SNMP
TCP 179	BGP
TCP 443	HTTPS
UDP/TCP 514	Syslog
UDP 520	RIP
UDP 546	DHCPv6 client
UDP 547	DHCPv6 server
UDP 2000	SCCP (Skinny)
TCP 3389	MS-RDP
UDP/TCP 5060	SIP

Les hôtes utilisent les ports TCP ou UDP pour identifier les sessions à l'origine (port source) et à la destination.

Par exemple, pour établir une session HTTP, l'hôte utilisera un port local au-delà de TCP1024 et le port TCP80 en destination.

Les numéros de ports par défaut des services applicatifs sont gérés par l'IANA.

3. Protocole TCP

Le texte qui suit sur le protocole TCP est directement inspiré de la page [Wikipedia sur TCP](#). Ce propos suffit amplement à maîtriser les objectifs de la certification CCNA.

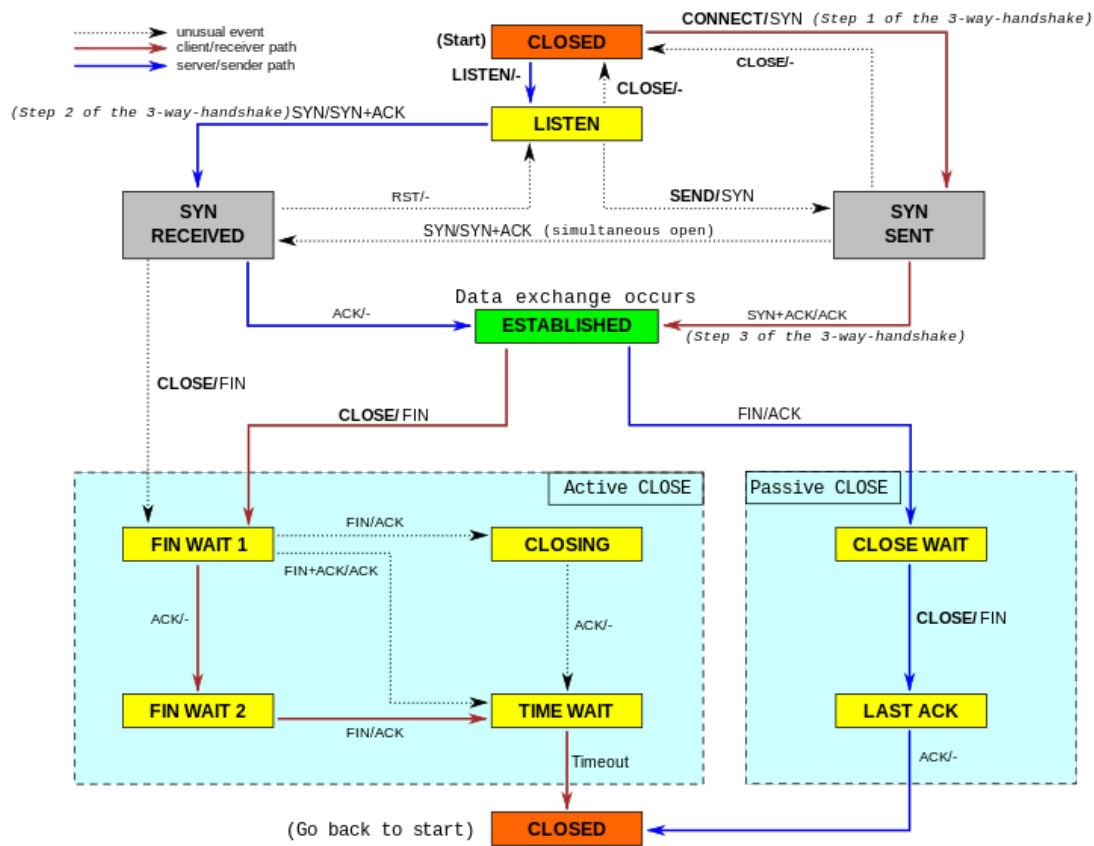
3.1. TCP : fonctionnement

Une session TCP fonctionne en trois phases :

- l'établissement de la connexion ;
- les transferts de données ;
- la fin de la connexion.

L'établissement de la connexion se fait par un *handshaking* en trois temps. La rupture de connexion, elle, utilise un *handshaking* en quatre temps. Pendant la phase d'établissement de la connexion, des paramètres comme le numéro de séquence sont initialisés afin d'assurer la transmission fiable (sans perte et dans l'ordre) des données.

3.2. Machine à état TCP



Machine à état TCP

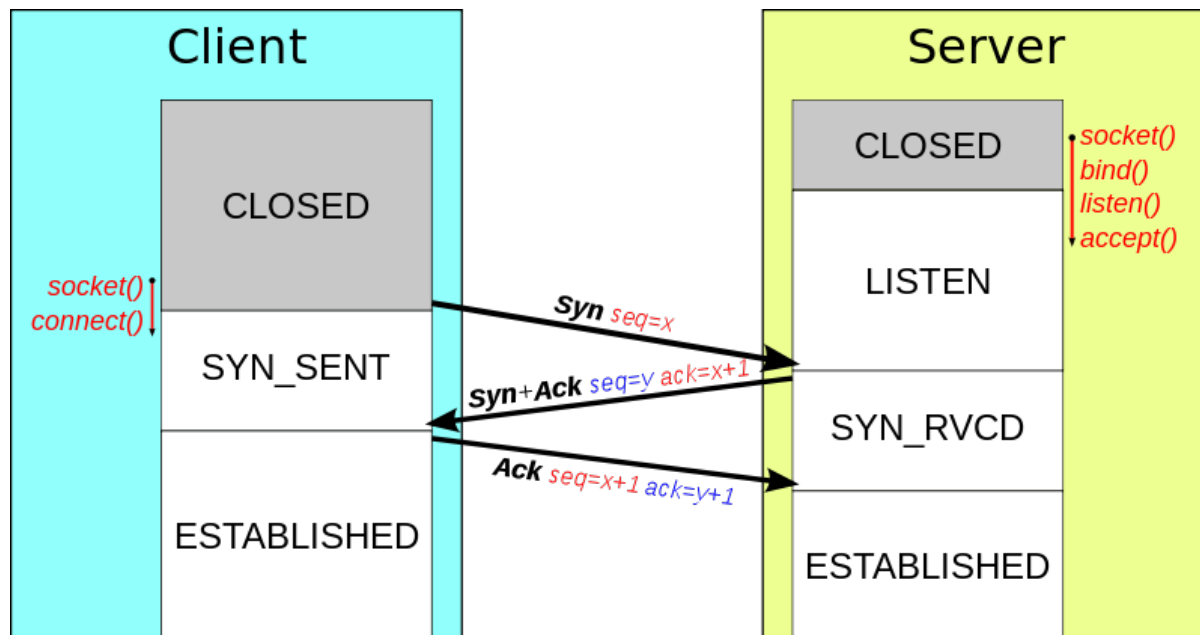
Source de l'image.

3.3. TCP Three Way Handshake

Même s'il est possible pour deux systèmes d'établir une connexion entre eux simultanément, dans le cas général, un système ouvre une 'socket' (point d'accès à une connexion TCP) et se met en attente passive de demandes de connexion d'un autre système. Ce fonctionnement est communément appelé ouverture passive, et est utilisé par le côté serveur de la connexion.

Le côté client de la connexion effectuée une ouverture active en 3 temps :

- Le client envoie un segment SYN au serveur,
- Le serveur lui répond par un segment SYN/ACK,
- Le client confirme par un segment ACK.



Connexion TCP

Source de l'image.

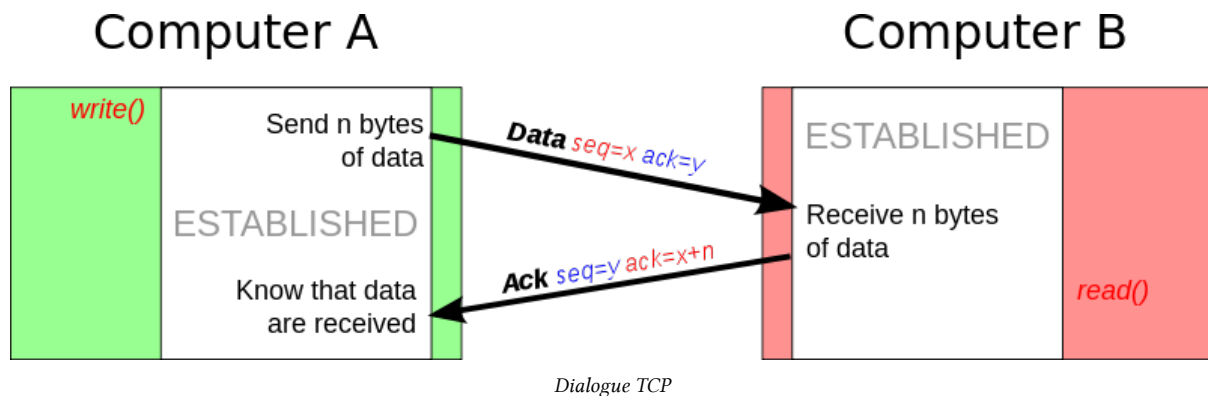
Durant cet échange initial, les numéros de séquence des deux parties sont synchronisés :

- Le client utilise son numéro de séquence initial dans le champ "Numéro de séquence" du segment SYN (x par exemple),
- Le serveur utilise son numéro de séquence initial dans le champ "Numéro de séquence" du segment SYN/ACK (y par exemple) et ajoute le numéro de séquence du client plus un (x+1) dans le champ "Numéro d'acquittement" du segment,
- Le client confirme en envoyant un ACK avec un numéro de séquence augmenté de un (x+1) et un numéro d'acquittement correspondant au numéro de séquence du serveur plus un (y+1).[^source-wikipedia]

3.4. Transfert des données

Pendant la phase de transferts de données, certains mécanismes clefs permettent d'assurer la robustesse et la fiabilité de TCP. En particulier :

- les numéros de séquence sont utilisés afin d'ordonner les segments TCP reçus et de détecter les données perdues,
- les sommes de contrôle permettent la détection d'erreurs,
- et les acquittements ainsi que les temporisations permettent la détection des segments perdus ou retardés.[^source-wikipedia]



Source de l'image.

3.5. Numéro de séquence et numéro d'acquittement

Le numéro d'acquittement est le numéro de séquence attendu du partenaire de communication.

ACK+1 signifie j'ai reçu les premiers segments donne-moi la suite.

On parle de numéro d'acquittement anticipatif.

3.6. Temporisation

La perte d'un segment est gérée par TCP en utilisant un mécanisme de temporisation et de retransmission. Après l'envoi d'un segment, TCP va attendre un certain temps la réception du ACK correspondant. Un temps trop court entraîne un grand nombre de retransmissions inutiles et un temps trop long ralentit la réaction en cas de perte d'un segment.

Dans les faits, le délai avant retransmission doit être supérieur au RTT moyen d'un segment, c'est-à-dire au temps que prend un segment pour effectuer l'aller-retour entre le client et le serveur. Comme cette valeur peut varier dans le temps, l'ordinateur "prélève" des échantillons à intervalle régulier et on en calcule une moyenne pondérée.

3.7. Somme de contrôle

Une somme de contrôle sur 16 bits, constituée par le complément à un de la somme complémentée à un de tous les éléments d'un segment TCP (en-tête et données), est calculée par l'émetteur, et incluse dans le segment émis. Le destinataire recalcule la somme de contrôle du segment reçu, et si elle correspond à la somme de contrôle reçue, on considère que le segment a été reçu intact et sans erreur.

3.8. Contrôle de flux

Chaque partenaire dans une connexion TCP dispose d'un tampon de réception dont la taille n'est pas illimitée. Afin d'éviter qu'un hôte ne surcharge l'autre, TCP prévoit plusieurs mécanismes de contrôle de flux. Ainsi, chaque segment TCP contient la taille disponible dans le tampon de réception de l'hôte qui l'a envoyé. En réponse, l'hôte distant va limiter la taille de la fenêtre d'envoi afin de ne pas le surcharger. D'autres algorithmes comme Nagle ou Clark facilitent également le contrôle du flux.

3.9. Contrôle de congestion

La congestion intervient lorsque trop de sources tentent d'envoyer trop de données trop vite pour que le réseau soit capable de les transmettre. Ceci entraîne la perte de nombreux paquets et de longs délais.

Les acquittements des données émises, ou l'absence d'acquittements, sont utilisés par les émetteurs pour interpréter de façon implicite l'état du réseau entre les systèmes finaux. À l'aide de temporisations, les émetteurs et destinataires TCP peuvent modifier le comportement du flux de données. C'est ce qu'on appelle généralement le contrôle de congestion.

Il existe une multitude d'algorithmes d'évitement de congestion pour TCP.

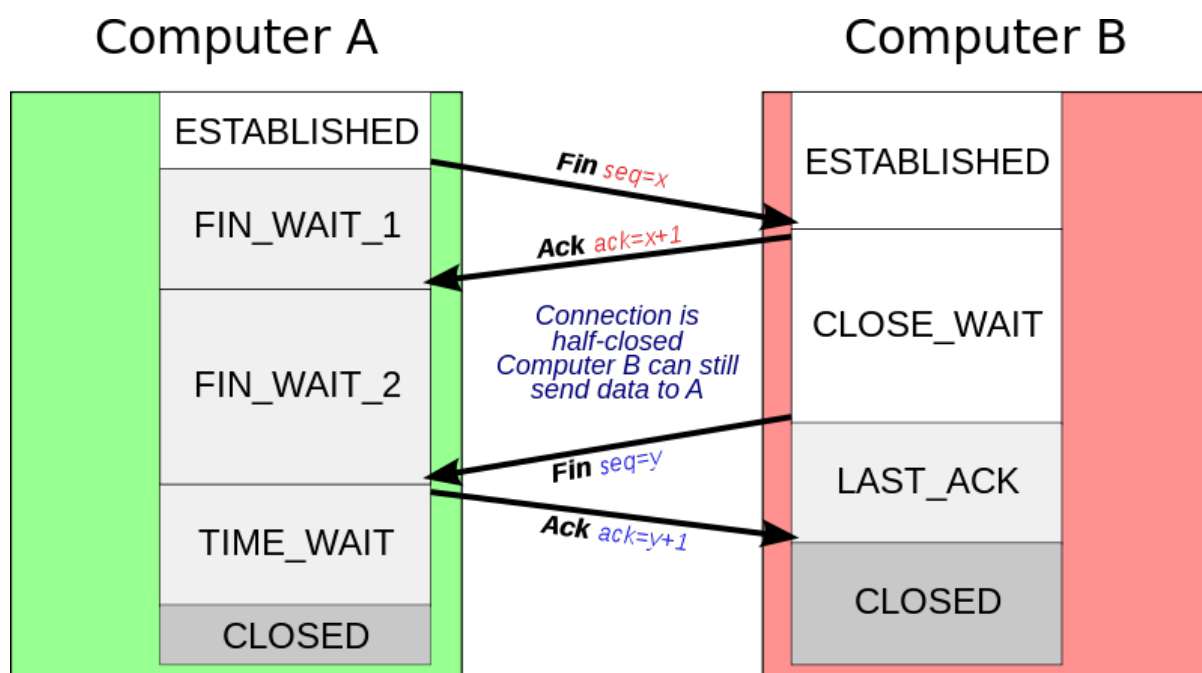
3.10. Autres fonctionnalités TCP

TCP utilise un certain nombre de mécanismes afin d'obtenir une bonne robustesse et des performances élevées. Ces mécanismes comprennent l'utilisation d'une fenêtre glissante, l'algorithme de démarrage lent (slow start), l'algorithme d'évitement de congestion (congestion avoidance), les algorithmes de retransmission rapide (fast retransmit) et de récupération rapide (fast recovery), etc.

Des recherches sont menées actuellement afin d'améliorer TCP pour traiter efficacement les pertes, minimiser les erreurs, gérer la congestion et être rapide dans des environnements très haut débit.

3.11. Fin d'une connexion

La phase de terminaison d'une connexion utilise un *handshaking* en quatre temps, chaque extrémité de la connexion effectuant sa terminaison de manière indépendante. Ainsi, la fin d'une connexion nécessite une paire de segments FIN et ACK pour chaque extrémité.



Fin de connexion TCP

Source de l'image.

4. Exemple de trafic TCP et UDP

Voici un exemple de trafic TCP et UDP obtenu à partir de la commande :

```
wget http://cisco.goffinet.org/logo.png
```

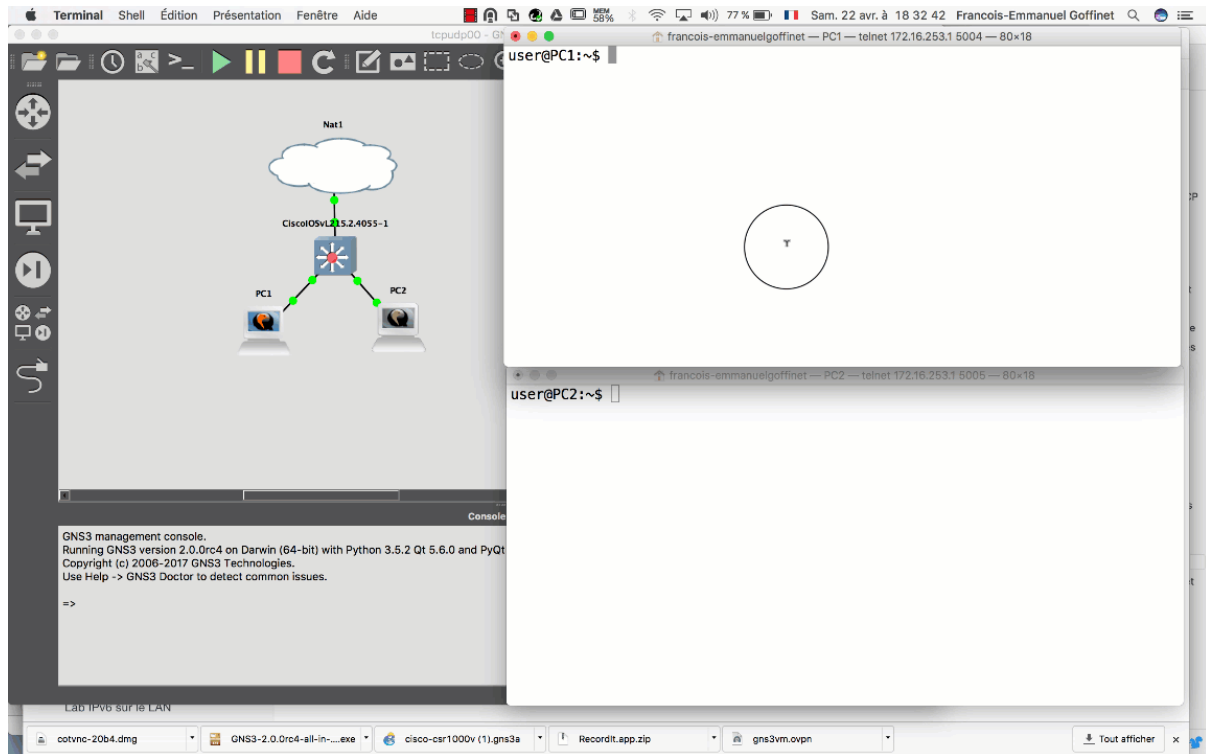
- Résolution DNS (UDP)
- Trafic HTTP (TCP)

On apprendra à distinguer :

- les numéros de ports source et destination

- les numéros de séquence et d'acquittement
- les trois étapes établissement, maintien et fin d'une connexion

Voici un autre exemple avec l'utilitaire Netcat :



Démonstration Netcat

5. Sources

- https://fr.wikipedia.org/wiki/Transmission_Control_Protocol
- https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- https://commons.wikimedia.org/wiki/File:Tcp_state_diagram_fixed_new.svg
- https://commons.wikimedia.org/wiki/File:Tcp_connect.svg
- https://commons.wikimedia.org/wiki/File:Tcp_talk.svg
- https://commons.wikimedia.org/wiki/File:Tcp_close.svg
- <http://nmap.org/book/tcpip-ref.html>

17. Lab vérifications et analyses TCP/IP

1. Objectifs

Cette activité consiste à déployer une topologie LAN simple connecté à l’Internet avec la configuration d’un routeur en Cisco IOS. C’est un aussi un “Lab de démarrage”. L’objectif principal est d’observer du trafic ICMP, TCP et UDP et de réaliser des tâches de diagnostic fondamental.

1.1. Préalable

Au préalable, on ira lire les documents qui exposent la nature de l’IOS Cisco, celui qui permet de mettre en place un environnement d’exercice avec GNS3, celui qui expose les méthodes de connexion à un routeur Cisco et enfin, une initiation à l’environnement en ligne de commande Cisco IOS.

- [Cisco IOS](#)
- [Installer et configurer GNS3](#)
- [Connexion à un routeur Cisco](#)
- [Méthode Cisco IOS CLI](#)

2. Construire la topologie



Topologie CCNALab01 : simple LAN connecté à l’Internet

2.1. Composants

Composant	Nom	Image	Rôle
Routeur	gateway	vios-adventerprisek9-m.vmdk.SPA.156-2.T	Passerelle IPv4 entre l’Internet et le LAN avec services DHCP, DHCPv6, SLAAC, DNS Forwarder
Nuage	Internet	Intégré au logiciel	Simule un Internet IPv4
Commutateur	SW0	Intégré au logiciel	Commutateur du LAN
Ordinateur	PC1	centos7.qcow2	PC client TCP/IP dans le LAN
Ordinateur	PC2	centos7.qcow2	PC client TCP/IP dans le LAN

2.2. Connexions

Périphérique 1	Interface	Interface	Périphérique 2	Réseau partagé
Internet	nat0	G0/1	-	-
gateway	G0/1	nat0	Internet	WAN
gateway	G0/0	e0	SW0	LAN
SW0	e0	G0/0	Gateway	LAN
SW0	e1	eth0	PC1	LAN
SW0	e2	eth0	PC2	LAN

2.3. Plan d'adressage

Une seule interface dispose de paramètres IPv4 et IPv6 statiques :

- G0/0 du routeur “gateway” :
- Adresse IPv4 privée : 192.168.1.254 255.255.255.0
- Adresse IPv6 Link-Local : fe80::1/64
- Adresse IPv6 privée : fd00:192:168:1::/64

Un service DHCP distribue les adresses dans cette plage de 192.168.1.1 à 192.168.1.199.

Le routeur s'annonce en DHCP comme passerelle et comme résolveur de noms DNS.

Le routeur remplit les tâches NAT qui interconnecte le réseau IPv4 privé 192.168.1.0/24 à l'Internet (public).

2.4. Configuration du routeur

Pour appliquer la configuration suivante, veuillez :

- Coller la configuration suivante dans la mémoire tampon
- Vous connecter à la console physique avec un logiciel de terminal.
- Copier le texte à partir du mode privilège avec l'invite router#

On remarque une configuration très sémantique.

```
configure terminal
!
hostname gateway
!
interface GigabitEthernet0/0
description LAN interface
ip address 192.168.1.254 255.255.255.0
ip nat inside
no shutdown
!
interface GigabitEthernet0/1
description WAN interface
ip address dhcp
ip nat outside
no shutdown
!
ip access-list standard LAN
permit 192.168.1.0 0.0.0.255
!
ip nat inside source list LAN interface GigabitEthernet0/1 overload
!
ip domain lookup
ip name-server 8.8.8.8
ip dns server
!
ip dhcp pool DHCP-LAN
network 192.168.1.0 255.255.255.0
default-router 192.168.1.254
```



```

dns-server 192.168.1.254
!
ip dhcp excluded-address 192.168.1.200 192.168.1.254
!
ipv6 unicast-routing
!
ipv6 dhcp pool DHCPv6-LAN
address prefix FD00:192:168:1::/64
dns-server FD00:192:168:1::1
!
interface GigabitEthernet0/0
ipv6 address FE80::1 link-local
ipv6 address FD00:192:168:1::1/64
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 dhcp server DHCPv6-LAN
!
interface GigabitEthernet0/1
description WAN interface
ipv6 address autoconfig
ipv6 nd ra suppress
!
end
!
wr

```

3. Vérifications à partir des clients TCP/IP

3.1. Paramètres IPv4

Visualisation de paramètres IPv4 :

```
[root@pc1 ~]* ip -4 add show dev eth0
```

```

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    inet 192.168.1.2/24 brd 192.168.1.255 scope global dynamic eth0
        valid_lft 83942sec preferred_lft 83942sec

```

Table de routage IPv4 :

```
[root@pc1 ~]* ip route
```

```

default via 192.168.1.254 dev eth0 proto static metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.2 metric 100

```

Résolveurs de noms :

```
[root@pc1 ~]* cat /etc/resolv.conf
```

```
# Generated by NetworkManager
nameserver 192.168.1.254
nameserver fd00:192:168:1::1
```

Test de connectivité locale (passerelle) :

```
[root@pc1 ~]# ping -c 1 192.168.1.254
```

```
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
64 bytes from 192.168.1.254: icmp_seq=1 ttl=255 time=1.38 ms

--- 192.168.1.254 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.389/1.389/1.389/0.000 ms
```

Test de connectivité globale :

```
[root@pc1 ~]# ping -c 1 8.8.8.8
```

```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=3.21 ms

--- 8.8.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.217/3.217/3.217/0.000 ms
```

Test de connectivité globale avec résolution de nom :

```
[root@pc1 ~]# ping -c 1 www.google.com
```

```
PING www.google.com (216.58.213.164) 56(84) bytes of data.
64 bytes from www.google.com (216.58.213.164): icmp_seq=1 ttl=52 time=2.24 ms

--- www.google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.247/2.247/2.247/0.000 ms
```

Test de résolution de noms locale :

```
user@ubuntu1604:~$ dig www.google.com
```

```
[root@pc1 ~]# dig www.google.com
```

```
; <<>> DiG 9.9.4-RedHat-9.9.4-38.el7_3.3 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39810
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                267     IN      A      216.58.213.164

;; Query time: 17 msec
;; SERVER: 192.168.1.254#53(192.168.1.254)
;; WHEN: Mon Jun 19 18:07:14 CEST 2017
;; MSG SIZE rcvd: 48
```

Test de résolution de noms distante :

```
[root@pc1 ~]* dig @8.8.8.8 www.google.com
```

```
; <<>> DiG 9.9.4-RedHat-9.9.4-38.el7_3.3 <<>> @8.8.8.8 www.google.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42683
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.                299     IN      A      216.58.213.164

;; Query time: 22 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Mon Jun 19 18:08:04 CEST 2017
;; MSG SIZE rcvd: 59
```

Connectivité HTTP :

```
curl ipinfo.io/ip
```

```
XX.XX.XX.XX
```

Table ARP avec de joindre une nouvelle adresse :

```
[root@pc1 ~]* ip -4 neighbor
```

```
192.168.1.254 dev eth0 lladdr 00:60:0d:08:b4:00 STALE
```

Table ARP après avoir joint une nouvelle adresse :

```
[root@pc1 ~]# ping -c 1 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=2.90 ms

--- 192.168.1.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.909/2.909/2.909/0.000 ms
```

```
[[root@pc1 ~]# ip -4 neighbor
```

```
192.168.1.1 dev eth0 lladdr 00:60:0d:05:04:00 STALE
192.168.1.254 dev eth0 lladdr 00:60:0d:08:b4:00 STALE
```

4. Configuration manuelle d'un hôte terminal

Tâches à exécuter avec les droits du super-utilisateur (root).

Arrêt du gestionnaire réseau :

```
systemctl stop networking || systemctl stop NetworkManager
```

Définition de l'adresse IPv4 et de son masque :

```
ip add add dev eth0 192.168.1.1/24
```

Activation de l'interface :

```
ip link set dev eth0 up
```

Création d'une route par défaut :

```
ip route add default via 192.168.1.254
```

Paramètre de résolveur de noms :

```
echo "nameserver 192.168.1.254" >> /etc/resolv.conf
```

5. Analyse de trafic

- Capture de trafic
- Génération de trafic
- Analyse de trafic

5.1. Trafic HTTP et DNS

Lancer une capture à partir de l'interface de PC1.

```
curl -i -X GET http://www.test.tf
curl -i -X HEAD http://www.test.tf
```

Sauvegarder la capture : <https://www.cloudshark.org/captures/70eea568e03c>.

5.2. Trafic TCP / UDP

Netcat est un outil libre de bas niveau qui sert à monter des tunnels TCP ou UDP. Seul, amélioré ou combiné avec des outils de chiffrement, il est une véritable boîte à outils du plombier du réseau.

On le propose ici dans un contexte client / serveur où PC1 joue le rôle de serveur TCP/UDP et PC2 joue le rôle de client TCP/UDP. L'objectif est de transmettre un dialogue texte du type "ici PC2", "ici PC1", "Au revoir" en UDP et en TCP.

Lancement du serveur TCP sur le port 4444 sur PC1 :

```
user@PC1:~$ nc -l -p 4444
```

Le client PC2 lance une session TCP sur le port 4444 de PC1 :

```
user@PC2:~$ nc 192.168.1.1 4444
ici PC2
```

Sur PC1, le message s'affiche et on répond :

```
user@PC1:~$ nc -l -p 4444
ici PC2
ici PC1
```

PC2 clôture la conversion :

```
user@PC2:~$ nc 192.168.1.1 4444
ici PC2
ici PC1
Au revoir
^C
```

Sauvegarder la capture : <https://www.cloudshark.org/captures/f54d3f43824a>.

Idem en UDP :

```
user@PC1:~$ nc -l -u -p 4444
```

```
user@PC2:~$ nc -u 192.168.1.1 4444
```

Sauvegarder la capture : <https://www.cloudshark.org/captures/2c9e082671c9>.

Quatrième partie Adressage IPv6

IPv6 est un sujet fortement vérifié dans les certifications Cisco CCNA. Cette partie s'intéresse à la reconnaissance et à la validation des adresses IPv6, à leur configuration sur les interfaces, à leur vérification et à leur diagnostic. Enfin, on trouvera un propos sur la manière de concevoir des plans d'adressage IPv6.

L'adressage IPv6 est codé en hexadécimal. Une telle étendue sur 128 bits nécessitait une représentation plus efficace que celle d'IPv4 (décimale pointée). Toutefois, l'usage de l'hexadécimal peut être un frein à l'acceptation et à la compréhension du protocole auprès de certains. Il s'agit d'une résistance que l'on peut facilement dépasser si on s'y intéresse le temps de la lecture de ces quelques articles.

On apprendra à identifier, à écrire et à vérifier des adresses : les adresses Unicast et Multicast et parmi elles les adresses Loopback (Node-Local), Link-local, Global Unicast, Unique Local, Well-Know Multicast, et Solicited-Node Multicast. On apprendra aussi à distinguer le préfixe et l'identifiant d'interface et à envisager le contrôle sur la construction de l'adresse. Enfin, on ne manquera pas de démontrer la souplesse de l'adressage en matière de découpage de blocs d'adresses.

18. Introduction aux adresses IPv6

On commencera par définir ces identifiants IPv6 encore trop peu connus. Ensuite, on apprendra à écrire, à simplifier et à valider une adresse IPv6.

1. Terminologie IPv6

- Un **lien** (*link*) est le support physique (ou la facilité telle un tunnel) de communication entre deux noeuds au niveau de la couche 2 liaisons de données/accès réseau (technologies LAN/WAN).
- Deux **noeuds** sur le même lien sont voisins (neighbors).
- Une **interface** est l'attachement d'un noeud au lien.
- Une **adresse** est un identifiant pour une interface (Unicast) ou pour un ensemble d'interfaces (Multicast). Une interface peut avoir plusieurs adresses IPv6 et être inscrite dans plusieurs groupes Multicast.
- Un **préfixe** désigne l'appartenance à un domaine IPv6.
- Le masque donne l'étendue du domaine IP : il se note après l'adresse et une barre oblique / ("slash"). Il indique le nombre de bits fixes dans une adresse.

2. Définition et étendue d'une adresse IPv6

On rappellera utilement que le protocole IPv6 doit être activé volontairement sur les interfaces des routeurs Cisco contrairement aux systèmes d'exploitation courants comme Windows ou Linux pour lesquels IPv6 est activé par défaut.

Les adresses IPv6 sont des identifiants uniques d'interfaces :

- codés sur 128 bits
- et notés en hexadécimal en 8 mots de 16 bits (4 hexas) séparés par des ":".

2001:0db8:00f4:0845:ea82:0627:e202:24fe /64

16 bits 16 bits 16 bits 16 bits 16 bits 16 bits 16 bits 16 bits
65536 65536 65536 65536 65536 65536 65536 65536

Exemple d'adresse IPv6 Global Unicast

Le **masque** identifie la partie fixe d'une adresse qui correspond aussi au numéro de réseau de 64 bits (le préfixe). Le **préfixe** est l'élément commun à toutes les adresses d'une même plage (au sein d'un réseau).

Par exemple, pour l'adresse "Global Unicast" 2001:0db8:00f4:0845:ea82:0627:e202:24fe/64 dans son écriture extensive :

2001:0db8:00f4:0845:ea82:0627:e202:24fe/64

16b 16b 16b 16b 16b 16b 16b 16b

Préfixe Interface ID Masque

Le **préfixe** ou la première adresse de la plage est `2001:0db8:00f4:0845::/64`. Le `/64` correspond aux 64 premiers bits de l'adresse que l'on peut aisément compter.

- Contrairement à IPv4, cette première adresse peut être attribuée à une interface.
- Une adresse IPv6 configurée sur une interface dispose dans 99,999 % des cas d'un masque par défaut de 64 bits noté `/64`.
- *A priori*, plus de calculs de masque exotique. La consigne est contraire à celle d'IPv4 conservatrice : "Jusqu'au gaspillage, réalisez des connexions en IPv6 !"
- Les 64 derniers bits appelés *identifiants d'interface* identifient l'interface dans le réseau.

À partir de l'exemple précédent, la valeur `ea82:0627:e202:24fe` correspond à l'identifiant d'interface.

La plage du bloc IPv6 s'étend sur 2 EXP 64 possibilités d'adresses, de la première adresse à la dernière :

```
2001:0db8:00f4:0845:0000:0000:0000:0000/64
2001:0db8:00f4:0845:ffff:ffff:ffff:ffff/64
-----
16b  16b  16b  16b  16b  16b  16b  16b
-----
Préfixe      Interface ID      Masque
```

Les 64 premiers bits sont identiques pour toutes les adresses d'un même bloc (`2001:0db8:00f4:0845`). Les 64 derniers bits varient de `0000:0000:0000:0000` à `ffff:ffff:ffff:ffff`. Cette dernière partie de 64 bits est appelée **Identifiant d'interface (IID Interface Identifier)**. Elle sert à identifier les interfaces au sein du réseau.

3. Ecriture résumée

Heureusement, on peut résumer l'écriture d'une adresse IPv6. Les systèmes sont obligés d'accepter ces simplifications d'écriture.

Par exemple, pour l'adresse "Global Unicast" `2001:0db8:00f4:0845:ea82:0627:e202:24fe/64` dans son écriture extensive :

```
2001:0db8:00f4:0845:ea82:0627:e202:24fe
-----
16b  16b  16b  16b  16b  16b  16b  16b
-----
Préfixe      Interface ID
```

Voici l'écriture résumée qui respecte deux règles faciles à retenir :

- Les zéros en en-tête de chaque mot peuvent être optimisés ;
- Une seule fois seulement, une suite consécutive de mots tout-à-zéro peut être résumée par `::`.

```
2001:0db8:00f4:0845:ea82:0627:e202:24fe
```

```
2001:-db8:-f4:-845:ea82:-627:e202:24fe
```


2001:db8:f4:845:ea82:627:e202:24fe

Ou encore l'adresse fe80:0000:0000:0000:bb38:9f98:0241:8a95 peut être résumée en fe80::bb38:9f98:241:8a95 :

fe80:0000:0000:0000:bb38:9f98:0241:8a95

fe80:----:----:----:bb38:9f98:-241:8a95

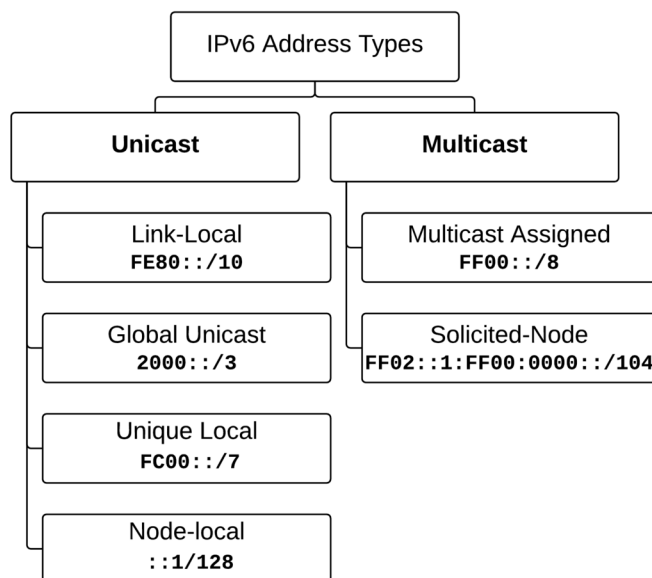
fe80::bb38:9f98:241:8a95

4. Types d'adresse IPv6

En IPv6, il y a pas mal de nouveautés en matière d'adresses :

- Le Broadcast disparaît d'IPv6. Il est remplacé par l'adressage Multicast plus fin.
- Une adresse préexiste toujours sur chaque interface activée en IPv6.
- Les adresses IPv6 rétablissent la connectivité globale (publique).
- Le NAT en IPv6 ne semble pas indispensable. Il est même déconseillé.
- Des adresses privées IPv6 restent néanmoins utiles pour transporter du trafic privé (sur des lignes privées, dans des tunnels VPN)

On trouvera plusieurs types d'adresse IPv6.



Types d'adresse IPv6

On distingue principalement les adresses **Unicast** des adresses **Multicast**.

Le RFC intitulé [IP Version 6 Addressing Architecture \(RFC 4291\)](#) indique les adresses dans lesquelles un hôte IPv6 doit se reconnaître :

- Une adresse Link-local sur chaque interface.
- Des adresses Unicast ou Anycast qui ont été configurées sur les interfaces du noeud.
- L'adresse de Loopback.
- Les adresses All-Nodes Multicast.
- L'adresse Solicited-Node Multicast pour chacune des adresses Unicast ou Anycast.
- Les adresses Multicast des groupes joints par le noeud.

Adresses Unicast

Une adresse Unicast est une adresse qui désigne une seule destination.

Parmi les adresses IPv6 Unicast, on trouve plusieurs classes réservées à certains usages.

- **Node-Local** `::1/128` : adresse de bouclage.
- **Link-Local** `fe80::/10` : adresse obligatoire et indispensable au bon fonctionnement du protocole.
- **Global Unicast** `2000::/3` : adresse publique.
- **Unique Local** `fc00::/7` (c'est le préfixe `fd00::/8` qui est utilisé) : adresse privée (aléatoire).

Une adresse **Anycast** ne se distingue pas d'une adresse Unicast ; ce terme "Anycast" désigne une adresse de livraison qui atteindra sa destination par une technique de routage au plus proche ou au plus efficient.

On remarquera aussi : **Unspecified** `::/128` : route non spécifiée, et **Default Route** `::/0` : route par défaut.

Adresses Multicast

Une adresse Multicast est une adresse qui désigne potentiellement plusieurs destinations. Les adresses Multicast remplacent utilement les adresses de Broadcast.

On trouvera deux types d'adresse Multicast que l'on reconnaît par leur préfixe `FF00::/8` :

- **Well-Know Multicast** : adresses Multicast bien connues
- **Solicited-Node Multicast** : adresses utilisées dans la découverte de voisinage (ND)

Concrètement, plusieurs interface du réseau et de l'inter-réseau se mettent à l'écoute d'une adresse Multicast, c'est-à-dire qu'elles seront potentiellement plusieurs à accepter du trafic à destination de cette adresse Multicast (si il leur est livré par les commutateurs).

5. Méthodes de configuration des interfaces

Une adresse IPv6 attribuée à une interface est constituée d'un préfixe de 64 bits et d'un identifiant d'interface de 64 bits.

Un identifiant d'interface peut être créé de différentes manières :

- statiquement : `2001:db8:14d6:1::1/64`, `2001:db8:14d6:1::254/64`, `fe80::101`, par exemple.
- par autoconfiguration (SLAAC) en utilisant l'une de ces trois méthodes :
 1. MAC EUI-64, par défaut ([RFC 4291](#))

2. tirage pseudo-aléatoire, par défaut chez Microsoft, Ubuntu, Mac OSX ([RFC 4941](#))
 3. CGA, peu implémenté ([RFC 3972](#))
- dynamiquement : DHCPv6 ([RFC 3315](#)) stateful, si le client est installé et activé (par défaut sur Microsoft Windows et Mac OSX)

Une interface IPv6 peut accepter plusieurs adresses dans un même préfixe mais aussi dans des préfixes distincts. L'idée est d'améliorer finement les politiques de routage et de filtrage en fonction de ces adresses.

6. Autoconfiguration Automatique

Comment les adresses parviennent-elles à se configurer toutes seules ? Le routeur du réseau local envoie régulièrement des messages "Neighbor Discovery" (ND) (ICMPv6 type 134), appelés des "Router Advertisements" qui poussent ces paramètres de configuration auprès des machines du réseau local. C'est le routeur IPv6 qui configure le réseau et de manière très fine. Les interfaces s'autoconfigurent selon la méthode SLAAC (*Stateless Address Autoconfiguration*) qui implique un trafic "Neighbor Discovery" (ND) de vérification avec l'usage des adresses Link-Local et du Multicast (NUD et DAD).

Si le Broadcast disparaît, ARP est remplacé par ND (ICMPv6). Ce n'est pas anodin et cette nouveauté mérite attention.

DHCPv6 est un nouveau protocole qui se décline en deux formules (avec et sans état) et qui peut se combiner à la méthode SLAAC d'autoconfiguration brièvement décrite ci-dessus. Cette dernière est juste activée par défaut sur tout hôte IPv6. L'usage de DHCPv6 nécessite un logiciel client DHCPv6 (déjà présent sous Windows et Mac) sur l'ordinateur client. Il est conseillé d'utiliser pleinement DHCPv6 dans un réseau bien géré. Il n'y a aucune obligation à laisser les hôtes terminaux s'autoconfigurer.

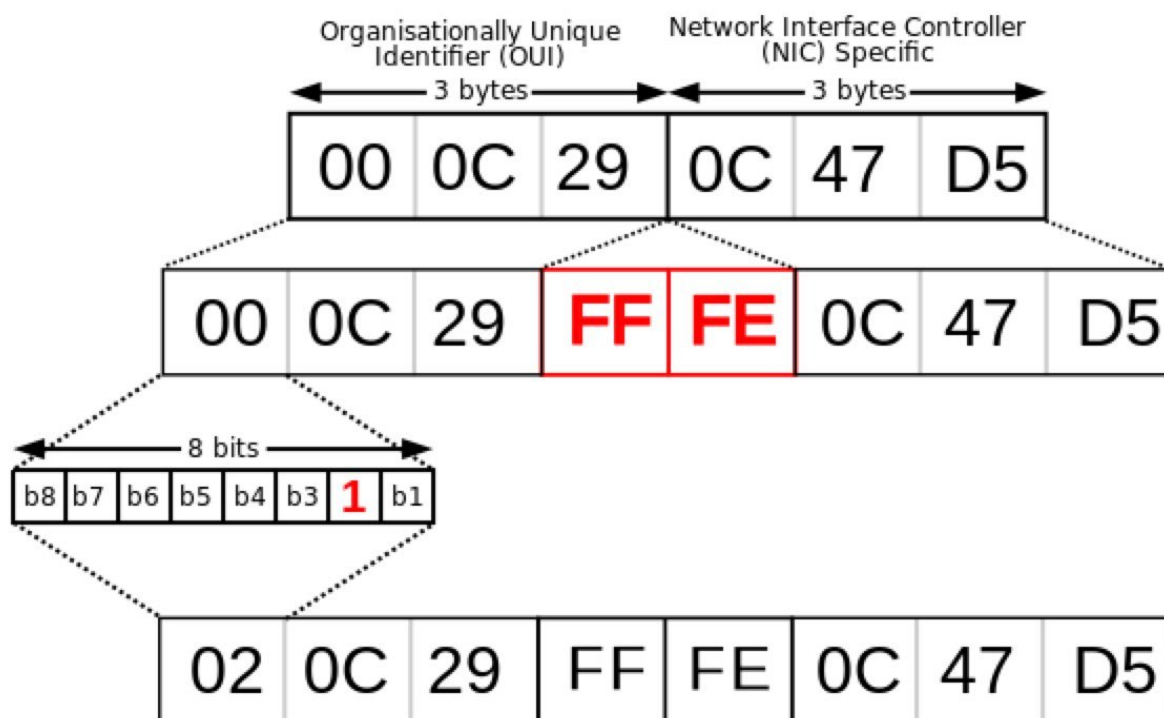
Enfin, c'est sans parler des comportements et configurations particulières sur Cisco IOS ou autres ou encore des stratégies et techniques de transition, DNS, la sécurité, le développement de code avec de l'IPv6.

7. Autoconfiguration des identifiants d'interface

Identifiant d'interface MAC-EUI64 "Modified"

MAC EUI-64 est une des méthodes de configuration automatique des ID d'interface qui se fonde sur l'adresse MAC IEEE 802 (48 bits).

On passe d'une adresse codée sur 48 bits à 64 bits en insérant 16 FF :FE au milieu de l'adresse MAC entre les 24 premiers bits et les 24 derniers bits. De plus, la méthode exige que le 7^{ième} bits de poids fort de l'adresse soit inversé passant de la valeur "Local" à "Universal" d'où ce terme "EUI64 Modified".



MAC-EUI 64 Modified

Pour compléter le propos, on ne manquera pas d'observer les paramètres acquis par une station Linux Debian 7 qui offre un service plus "standard" dans les mêmes conditions :

```
root@debian:~# ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:59:70:f3
          inet addr:192.168.1.119 Bcast:192.168.1.255 Masque:255.255.255.0
          adr inet6: fd26:44e1:8c70:fd00:ba27:ebff:fe59:70f3/64 Scope:Global
          adr inet6: fe80::ba27:ebff:fe59:70f3/64 Scope:Lien
          adr inet6: 2001:db8:acaf:fd00:ba27:ebff:fe59:70f3/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:93310 errors:0 dropped:49 overruns:0 frame:0
          TX packets:76990 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:11548535 (11.0 MiB)  TX bytes:14021689 (13.3 MiB)

lo        Link encap:Boucle locale
          inet addr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:17546 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17546 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:1199358 (1.1 MiB)  TX bytes:1199358 (1.1 MiB)
```

On trouvera sur cet ordinateur deux interfaces `eth0` et `lo`. La sortie donne la portée de chaque adresse IPv6.

L'interface `eth0` connectée au réseau local ne démarre pas par défaut un client DHCPv6, ce n'est pas erreur. Sur cette interface, on trouvera les trois types d'adresses. Pour chacune, l'identifiant d'interface est identique. Il est généré par la méthode MAC-EUI64 (méthode par défaut). MAC-EUI64 est une méthode standardisée qui vise à obtenir un identifiant d'interface de 64 bits à partir d'une adresse MAC de 48 bits. Aussi, ce n'est pas une erreur mais le comportement par défaut.

```
fe80::ba27:ebff:fe59:70f3
2001:db8:acaf:fd00:ba27:ebff:fe59:70f3
fd26:44e1:8c70:fd00:ba27:ebff:fe59:70f3
```

Avez-vous aperçu le souci de confidentialité ? On retrouve l'adresse MAC de l'interface dans l'identifiant d'interface de cette adresse. Cette portion significative de l'adresse permet d'identifier de manière certaine l'interface quel que soit son préfixe. De plus, les 24 premiers bits de l'adresse MAC (ici b8:27:eb) identifient le matériel ; à vous de le retrouver.

Soit la méthode MAC-EUI64 insère 16 bits manquants FFFE après les 24 premiers bits de l'adresse MAC (OUI, *Organizationally Unique Identifier*) dont le 7e bit a été inversé :

de b8:27:eb :59:70:f3, on passe à ba27:ebff :fe59:70f3

16 bits ont été ajoutés et la valeur du deuxième hexa est passée de 8 à a soit augmenté de 1 bit. Ce renversement signifie que l'adresse est "administrée localement".

Il est possible de modifier ce comportement par défaut quel que soit le système d'exploitation.

Privacy Extensions for Stateless Address Autoconfiguration in IPv6

Alors que la méthode de configuration des interfaces par défaut obligatoire est la méthode MAC-EUI64, les hôtes Windows et Linux domestiques préfèrent la méthode "Privacy Extension" ([Privacy Extensions for Stateless Address Autoconfiguration in IPv6](#)).

Cette méthode autoconfigure deux identifiants d'interface pour une durée limitée dans chaque préfixe :

- Une adresse "publique" (c'est-à-dire non secrète) pour les serveurs, enregistrés dans un serveur DNS par exemple, qui est utilisée pour accepter des connexions directes entrantes venant d'autres machines. Des pare-deux dans le chemin pourraient bloquer ce trafic.
- Une adresse "temporary" utilisée comme "bouclier" sur l'identité des clients quand ils débutent une connexion.

On trouvera des exemples de ces adresses et identifiants d'interfaces autoconfigurés "Privacy Extensions" dans la page [Prise d'information IPv6](#).

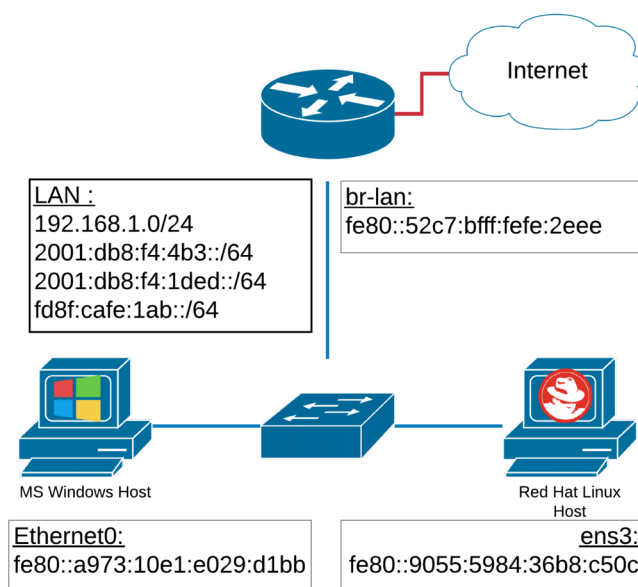
8. Résumé

Manque adresse loopback

Link-Local	Global Unicast	Unique Local
L'adressage Link-Local (Unicast) se reconnaît par : un préfixe FE80::/10 et un identifiant d'interface de 64 bits autoconfiguré (MAC-EUI64 ou aléatoire) ou fixé (Cisco)	Préfixe : 2000::/3 (2000::/4)	Préfixe : FC00::/7 (FD00::/8)
Est obligatoire sur toutes les interfaces quand IPv6 est activé	optionnel	optionnel
Ces destinations ne sont jamais transférées par les routeurs !	Connectivité de bout en bout nécessaire	fonctionne dans un interrèseau privé
Ces adresses sont utilisées dans le trafic de gestion comme ND, SLAAC, avec les protocoles de routage.	Trafic vers l'Internet	Trafic sur des lignes privées (physiques ou virtuelles).

19. Prise d'information IPv6

1. Topologie



Topologie Prise d'information IPv6

1.1. Présentation

Chaque interface connectée au réseau dispose d'une adresse Link-Local :

- Hôte Windows (Ethernet0) : fe80::a973:10e1:e029:d1bb%13
- Hôte Red Hat : fe80::9055:5984:36b8:c50c%ens3
- Routeur Linux : fe80::52c7:bfff:fefe:2eee%br-lan

On trouvera trois préfixes configurés sur le réseau :

- 2001:db8:f4:2b4::/64
- 2001:db8:f4:3bc::/64
- fd8f:cafe:1ab::/64

Selon les capacités des hôtes, chaque préfixe fera l'objet d'une autoconfiguration automatique d'adresses et d'une attribution dynamique (DHCPv6).

1.2. Objectifs de l'activité

On propose dans cet exercice pratique de visualiser les différentes tables IPv6 d'un hôte Windows et d'un hôte Linux.

2. Hôte Windows

2.1. Liste des commandes Windows

Trois outils natifs sont disponibles pour vérifier IPv6 sous Microsoft Windows : `ipconfig`, `netsh` et PowerShell.

- `ipconfig /all`
- `netsh interface ipv6 show interface`
- `netsh interface ipv6 show address`
- `netsh interface ipv6 show joins`
- `netsh interface ipv6 show neighbor`
- `netsh interface ipv6 show route`
- `Get-NetIPInterface -AddressFamily IPv6`
- `Get-NetIPAddress -AddressFamily IPv6`
- `Get-NetRoute -AddressFamily IPv6`
- `Get-NetNeighbor -AddressFamily IPv6`

2.2. `ipconfig /all`

Cette commande `ipconfig /all` est bien connue des administrateurs Windows mais elle donne des résultats sommaires pour IPv6.

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.
```

```
C:\Users\user>ipconfig /all
```

```
Windows IP Configuration
```

```
Host Name . . . . . : DESKTOP-N62C2SU
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : lan
```

```
Ethernet adapter Ethernet0:
```

```
Connection-specific DNS Suffix . : lan
Description . . . . . : Intel(R) 82574L Gigabit Network Connection
Physical Address. . . . . : 00-0C-29-0F-07-EC
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IPv6 Address. . . . . : 2001:db8:f4:2b4::d4d(Preferred)
Lease Obtained. . . . . : Wednesday, 6 February 2019 22:47:44
Lease Expires . . . . . : Thursday, 21 February 2019 01:25:27
IPv6 Address. . . . . : 2001:db8:f4:3bc::d4d(Preferred)
```

```

Lease Obtained. . . . . : Thursday, 7 February 2019 19:29:50
Lease Expires . . . . . : Tuesday, 19 February 2019 06:15:35
IPv6 Address. . . . . : 2001:db8:f4:3bc:a973:10e1:e029:d1bb(Preferred)
IPv6 Address. . . . . : fd8f:cafe:1ab::d4d(Preferred)
Lease Obtained. . . . . : Wednesday, 6 February 2019 22:47:45
Lease Expires . . . . . : Monday, 17 March 2155 01:59:26
IPv6 Address. . . . . : fd8f:cafe:1ab:0:a973:10e1:e029:d1bb(Preferred)
Temporary IPv6 Address. . . . . : 2001:db8:f4:3bc:7c80:3143:f2c8:8a05(Preferred)
Temporary IPv6 Address. . . . . : fd8f:cafe:1ab:0:7c80:3143:f2c8:8a05(Preferred)
Link-local IPv6 Address . . . . . : fe80::a973:10e1:e029:d1bb%13(Preferred)
IPv4 Address. . . . . : 192.168.1.171(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : Thursday, 7 February 2019 19:29:45
Lease Expires . . . . . : Friday, 8 February 2019 07:29:44
Default Gateway . . . . . : fe80::52c7:bfff:fefe:2eee%13
                          192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 67111977
DHCPv6 Client DUID. . . . . : 00-01-00-01-22-50-08-0B-00-0C-29-0F-07-EC
DNS Servers . . . . . : fd8f:cafe:1ab::1
                          192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled

```

Ethernet adapter Npcap Loopback Adapter:

```

Connection-specific DNS Suffix . :
Description . . . . . : Npcap Loopback Adapter
Physical Address. . . . . : 02-00-4C-4F-4F-50
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::bcfb:20de:f90b:d525%14(Preferred)
Autoconfiguration IPv4 Address. . : 169.254.213.37(Preferred)
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 486670412
DHCPv6 Client DUID. . . . . : 00-01-00-01-22-50-08-0B-00-0C-29-0F-07-EC
DNS Servers . . . . . : fec0:0:0:ffff::1%1
                          fec0:0:0:ffff::2%1
                          fec0:0:0:ffff::3%1
NetBIOS over Tcpip. . . . . : Enabled

```

Ethernet adapter Bluetooth Network Connection:

```

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Bluetooth Device (Personal Area Network)
Physical Address. . . . . : AC-BC-32-95-7F-5C
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes

```

Tunnel adapter Teredo Tunneling Pseudo-Interface:

```

Connection-specific DNS Suffix . :
Description . . . . . : Teredo Tunneling Pseudo-Interface
Physical Address. . . . . : 00-00-00-00-00-00-00-E0
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
IPv6 Address. . . . . : 2001:0:9d38:6ab8:77:3985:3f57:fe54(Preferred)

```



```

Link-local IPv6 Address . . . . . : fe80::77:3985:3f57:fe54%7(Preferred)
Default Gateway . . . . . :
DHCPv6 IAID . . . . . : 234881024
DHCPv6 Client DUID. . . . . : 00-01-00-01-22-50-08-0B-00-0C-29-0F-07-EC
NetBIOS over Tcpip. . . . . : Disabled

```

2.3. netsh interface ipv6 show interface

La commande `netsh interface ipv6 show interface` permet d'identifier les interface IPv6 dans Windows.

```

Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

```

```
C:\Users\user>netsh interface ipv6 show interface
```

Idx	Met	MTU	State	Name
1	75	4294967295	connected	Loopback Pseudo-Interface 1
7	75	1280	connected	Teredo Tunneling Pseudo-Interface
13	25	1500	connected	Ethernet0
4	65	1500	disconnected	Bluetooth Network Connection
14	25	1500	connected	Npcap Loopback Adapter

2.4. netsh interface ipv6 show address

La commande `netsh interface ipv6 show address` affiche les adresses par interface avec des détails leur statut et leur délai de validité.

```

Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

```

```
C:\Users\user>netsh interface ipv6 show address
```

```
Interface 1: Loopback Pseudo-Interface 1
```

Addr Type	DAD State	Valid Life	Pref. Life	Address
Other	Preferred	infinite	infinite	::1

```
Interface 7: Teredo Tunneling Pseudo-Interface
```

Addr Type	DAD State	Valid Life	Pref. Life	Address
Public	Preferred	infinite	infinite	2001:0:9d38:6ab8:77:3985:3f57:fe54
Other	Preferred	infinite	infinite	fe80::77:3985:3f57:fe54%7

```
Interface 13: Ethernet0
```

Addr Type	DAD State	Valid Life	Pref. Life	Address
Dhcp	Preferred	13d5h48m39s	6d5h48m39s	2001:db8:f4:2b4::d4d
Dhcp	Preferred	11d10h38m47s	6d23h47m5s	2001:db8:f4:3bc::d4d
Temporary	Preferred	6d23h52m58s	23h46m21s	2001:db8:f4:3bc:7c80:3143:f2c8:8a05
Public	Preferred	11d10h38m48s	6d23h47m6s	2001:db8:f4:3bc:a973:10e1:e029:d1bb
Dhcp	Preferred	infinite	infinite	fd8f:cafe:1ab::d4d

```
Temporary Preferred 6d23h52m58s 23h46m21s fd8f:cafe:1ab:0:7c80:3143:f2c8:8a05
Public Preferred infinite infinite fd8f:cafe:1ab:0:a973:10e1:e029:d1bb
Other Preferred infinite infinite fe80::a973:10e1:e029:d1bb%13
```

Interface 4: Bluetooth Network Connection

```
Addr Type DAD State Valid Life Pref. Life Address
-----
Other      Deprecated      infinite      infinite fe80::68c0:6c2e:fe7c:4c9b%4
```

Interface 14: Npcap Loopback Adapter

```
Addr Type DAD State Valid Life Pref. Life Address
-----
Other      Preferred      infinite      infinite fe80::bcfb:20de:f90b:d525%14
```

2.5. netsh interface ipv6 show address interface=X

La commande `netsh interface ipv6 show address interface=X` offre ces détails par adresse.

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.
```

```
C:\Users\user>netsh interface ipv6 show address interface=13
```

Address 2001:db8:f4:2b4::d4d Parameters

```
-----
Interface Luid      : Ethernet0
Scope Id            : 0.0
Valid Lifetime      : 13d5h47m33s
Preferred Lifetime  : 6d5h47m33s
DAD State           : Preferred
Address Type        : Dhcp
Skip as Source      : false
```

Address 2001:db8:f4:3bc::d4d Parameters

```
-----
Interface Luid      : Ethernet0
Scope Id            : 0.0
Valid Lifetime      : 11d10h37m41s
Preferred Lifetime  : 6d23h45m59s
DAD State           : Preferred
Address Type        : Dhcp
Skip as Source      : false
```

Address 2001:db8:f4:3bc:7c80:3143:f2c8:8a05 Parameters

```
-----
Interface Luid      : Ethernet0
Scope Id            : 0.0
Valid Lifetime      : 6d23h51m52s
Preferred Lifetime  : 23h45m15s
DAD State           : Preferred
Address Type        : Temporary
Skip as Source      : false
```

Address 2001:db8:f4:3bc:a973:10e1:e029:d1bb Parameters

```
-----
```

```
Interface Luid      : Ethernet0
Scope Id           : 0.0
Valid Lifetime     : 11d10h37m41s
Preferred Lifetime : 6d23h45m59s
DAD State          : Preferred
Address Type       : Public
Skip as Source     : false
```

Address fd8f:cafe:1ab::d4d Parameters

```
-----
Interface Luid      : Ethernet0
Scope Id           : 0.0
Valid Lifetime     : infinite
Preferred Lifetime : infinite
DAD State          : Preferred
Address Type       : Dhcp
Skip as Source     : false
```

Address fd8f:cafe:1ab:0:7c80:3143:f2c8:8a05 Parameters

```
-----
Interface Luid      : Ethernet0
Scope Id           : 0.0
Valid Lifetime     : 6d23h51m52s
Preferred Lifetime : 23h45m15s
DAD State          : Preferred
Address Type       : Temporary
Skip as Source     : false
```

Address fd8f:cafe:1ab:0:a973:10e1:e029:d1bb Parameters

```
-----
Interface Luid      : Ethernet0
Scope Id           : 0.0
Valid Lifetime     : infinite
Preferred Lifetime : infinite
DAD State          : Preferred
Address Type       : Public
Skip as Source     : false
```

Address fe80::a973:10e1:e029:d1bb%13 Parameters

```
-----
Interface Luid      : Ethernet0
Scope Id           : 0.13
Valid Lifetime     : infinite
Preferred Lifetime : infinite
DAD State          : Preferred
Address Type       : Other
Skip as Source     : false
```

2.6. netsh interface ipv6 show joins

La commande `netsh interface ipv6 show joins` indique par interface les groupes Multicast auxquelles elles sont à l'écoute.

Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\user>netsh interface ipv6 show joins

Interface 1: Loopback Pseudo-Interface 1

Scope	References	Last	Address
0	1	Yes	ff02::c

Interface 7: Teredo Tunneling Pseudo-Interface

Scope	References	Last	Address
0	0	Yes	ff01::1
0	0	Yes	ff02::1
0	2	Yes	ff02::1:ff57:fe54

Interface 13: Ethernet0

Scope	References	Last	Address
0	0	No	ff01::1
0	0	No	ff02::1
0	1	Yes	ff02::c
0	1	Yes	ff02::fb
0	1	Yes	ff02::1:3
0	3	Yes	ff02::1:ff00:d4d
0	3	Yes	ff02::1:ff29:d1bb
0	2	Yes	ff02::1:ffc8:8a05
0	0	No	ff02::1:ffe3:ecd8

Interface 4: Bluetooth Network Connection

Scope	References	Last	Address
0	0	Yes	ff01::1
0	0	Yes	ff02::1
0	1	Yes	ff02::1:ff7c:4c9b

Interface 14: Npcap Loopback Adapter

Scope	References	Last	Address
0	0	Yes	ff01::1
0	0	Yes	ff02::1
0	1	Yes	ff02::c
0	1	Yes	ff02::fb
0	1	Yes	ff02::1:3
0	1	Yes	ff02::1:ff0b:d525

2.7. netsh interface ipv6 show neighbor

Equivalente à la commande arp -a, la commande netsh interface ipv6 show neighbor donne la table de voisinage IPv6.

Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\user>netsh interface ipv6 show neighbor

netsh interface ipv6 show neighbor

Interface 1: Loopback Pseudo-Interface 1

Internet Address	Physical Address	Type
ff02::2		Permanent
ff02::c		Permanent
ff02::16		Permanent
ff02::1:2		Permanent
ff02::1:ff00:1		Permanent
ff05::c		Permanent
ff05::1:3		Permanent

Interface 7: Teredo Tunneling Pseudo-Interface

Internet Address	Physical Address	Type
2001:0:9d38:6ab8:ccd:209b:53eb:f5fc	255.255.255.255:65535	Unreachable
fe80::8000:f227:62c7:9547	255.255.255.255:65535	Unreachable (Router)
ff02::2	255.255.255.255:65535	Permanent
ff02::16	255.255.255.255:65535	Permanent
ff02::1:2	255.255.255.255:65535	Permanent
ff02::1:ff00:1	255.255.255.255:65535	Permanent
ff05::c	255.255.255.255:65535	Permanent
ff05::1:3	255.255.255.255:65535	Permanent

Interface 13: Ethernet0

Internet Address	Physical Address	Type
fd8f:cafe:1ab::1	50-c7-bf-fe-2e-ee	Stale (Router)
fe80::52c7:bfff:fe2e:2eee	50-c7-bf-fe-2e-ee	Stale (Router)
ff02::1	33-33-00-00-00-01	Permanent
ff02::2	33-33-00-00-00-02	Permanent
ff02::c	33-33-00-00-00-0c	Permanent
ff02::16	33-33-00-00-00-16	Permanent
ff02::fb	33-33-00-00-00-fb	Permanent
ff02::1:2	33-33-00-01-00-02	Permanent
ff02::1:3	33-33-00-01-00-03	Permanent
ff02::1:ff00:1	33-33-ff-00-00-01	Permanent
ff02::1:ff00:d4d	33-33-ff-00-0d-4d	Permanent
ff02::1:ff29:d1bb	33-33-ff-29-d1-bb	Permanent
ff02::1:ffc8:8a05	33-33-ff-c8-8a-05	Permanent
ff02::1:ffe3:ecd8	33-33-ff-e3-ec-d8	Permanent
ff02::1:fffe:2eee	33-33-ff-fe-2e-ee	Permanent
ff05::c	33-33-00-00-00-0c	Permanent
ff05::1:3	33-33-00-01-00-03	Permanent

Interface 4: Bluetooth Network Connection

Internet Address	Physical Address	Type
-----	-----	-----
ff02::16	33-33-00-00-00-16	Permanent
ff02::1:2	33-33-00-01-00-02	Permanent
ff02::1:ff00:1	33-33-ff-00-00-01	Permanent
ff05::c	33-33-00-00-00-0c	Permanent
ff05::1:3	33-33-00-01-00-03	Permanent

Interface 14: Npcap Loopback Adapter

Internet Address	Physical Address	Type
-----	-----	-----
ff02::1	33-33-00-00-00-01	Permanent
ff02::2	33-33-00-00-00-02	Permanent
ff02::c	33-33-00-00-00-0c	Permanent
ff02::16	33-33-00-00-00-16	Permanent
ff02::1:2	33-33-00-01-00-02	Permanent
ff02::1:3	33-33-00-01-00-03	Permanent
ff02::1:ff00:1	33-33-ff-00-00-01	Permanent
ff02::1:ff0b:d525	33-33-ff-0b-d5-25	Permanent
ff05::c	33-33-00-00-00-0c	Permanent
ff05::1:3	33-33-00-01-00-03	Permanent

2.8. netsh interface ipv6 show route

La commande `netsh interface ipv6 show route` donne la table de routage IPv6.

Microsoft Windows [Version 10.0.16299.309]

(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\user>netsh interface ipv6 show route

Publish	Type	Met	Prefix	Idx	Gateway/Interface Name
-----	-----	---	-----	---	-----
No	Manual	256	::/0	13	fe80::52c7:bfff:fefe:2eee
No	System	256	::1/128	1	Loopback Pseudo-Interface 1
No	Manual	256	2001::/32	7	Teredo Tunneling Pseudo-Interface
No	System	256	2001:0:9d38:6ab8:77:3985:3f57:fe54/128	7	Teredo Tunneling Pseudo-Interface
No	System	256	2001:db8:f4:2b4::d4d/128	13	Ethernet0
No	Manual	256	2001:db8:f4:3bc::/64	13	Ethernet0
No	Manual	256	2001:db8:f4:3bc::/64	13	fe80::52c7:bfff:fefe:2eee
No	System	256	2001:db8:f4:3bc::d4d/128	13	Ethernet0
No	System	256	2001:db8:f4:3bc:7c80:3143:f2c8:8a05/128	13	Ethernet0
No	System	256	2001:db8:f4:3bc:a973:10e1:e029:d1bb/128	13	Ethernet0
No	Manual	256	fd8f:cafe:1ab::/48	13	fe80::52c7:bfff:fefe:2eee
No	Manual	256	fd8f:cafe:1ab::/64	13	Ethernet0
No	System	256	fd8f:cafe:1ab::d4d/128	13	Ethernet0
No	System	256	fd8f:cafe:1ab:0:7c80:3143:f2c8:8a05/128	13	Ethernet0
No	System	256	fd8f:cafe:1ab:0:a973:10e1:e029:d1bb/128	13	Ethernet0
No	System	256	fe80::/64	14	Npcap Loopback Adapter
No	System	256	fe80::/64	13	Ethernet0
No	System	256	fe80::/64	4	Bluetooth Network Connection
No	System	256	fe80::/64	7	Teredo Tunneling Pseudo-Interface
No	System	256	fe80::77:3985:3f57:fe54/128	7	Teredo Tunneling Pseudo-Interface

```

No      System      256  fe80::68c0:6c2e:fe7c:4c9b/128    4  Bluetooth Network Connection
No      System      256  fe80::a973:10e1:e029:d1bb/128   13 Ethernet0
No      System      256  fe80::bcfb:20de:f90b:d525/128   14 Npcap Loopback Adapter
No      System      256  ff00::/8                          1  Loopback Pseudo-Interface 1
No      System      256  ff00::/8                          14 Npcap Loopback Adapter
No      System      256  ff00::/8                          13 Ethernet0
No      System      256  ff00::/8                          7  Teredo Tunneling Pseudo-Interface
No      System      256  ff00::/8                          4  Bluetooth Network Connection

```

2.9. Get-NetIPInterface -AddressFamily IPv6

La commande `Get-NetIPInterface -AddressFamily IPv6` correspond à la commande `netsh interface ipv6 show interface`.

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\user> Get-NetIPInterface -AddressFamily IPv6

ifIndex	InterfaceAlias	AddressFamily	NlMtu(Bytes)	InterfaceMetric	Dhcp	Connection\State	PolicyStore
14	Npcap Loopback Adapter	IPv6	1500	25	Enabled	Connected	\ActiveStore
4	Bluetooth Network Connection	IPv6	1500	65	Disabled	Disconnected	\ActiveStore
13	Ethernet0	IPv6	1500	25	Enabled	Connected	\ActiveStore
7	Teredo Tunneling Pseudo-Interface 1	IPv6	1280	75	Enabled	Connected	\ActiveStore
1	Loopback Pseudo-Interface 1	IPv6	4294967295	75	Disabled	Connected	\ActiveStore

2.10. Get-NetIPAddress -AddressFamily IPv6

La commande `Get-NetIPAddress -AddressFamily IPv6` correspond à la commande `netsh interface ipv6 show address interface`.

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\user> Get-NetIPAddress -AddressFamily IPv6

```

IPAddress      : fe80::bcfb:20de:f90b:d525%14
InterfaceIndex : 14
InterfaceAlias : Npcap Loopback Adapter
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 64
PrefixOrigin   : WellKnown
SuffixOrigin   : Link
AddressState   : Preferred
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)

```

```
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource      : False
PolicyStore       : ActiveStore
```

```
IPAddress        : fe80::68c0:6c2e:fe7c:4c9b%4
InterfaceIndex   : 4
InterfaceAlias    : Bluetooth Network Connection
AddressFamily     : IPv6
Type             : Unicast
PrefixLength     : 64
PrefixOrigin     : WellKnown
SuffixOrigin     : Link
AddressState     : Deprecated
ValidLifetime    : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource     : False
PolicyStore      : ActiveStore
```

```
IPAddress        : fe80::a973:10e1:e029:d1bb%13
InterfaceIndex   : 13
InterfaceAlias    : Ethernet0
AddressFamily     : IPv6
Type             : Unicast
PrefixLength     : 64
PrefixOrigin     : WellKnown
SuffixOrigin     : Link
AddressState     : Preferred
ValidLifetime    : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource     : False
PolicyStore      : ActiveStore
```

```
IPAddress        : fd8f:cafe:1ab:0:a973:10e1:e029:d1bb
InterfaceIndex   : 13
InterfaceAlias    : Ethernet0
AddressFamily     : IPv6
Type             : Unicast
PrefixLength     : 64
PrefixOrigin     : RouterAdvertisement
SuffixOrigin     : Link
AddressState     : Preferred
ValidLifetime    : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource     : False
PolicyStore      : ActiveStore
```

```
IPAddress        : fd8f:cafe:1ab:0:7c80:3143:f2c8:8a05
InterfaceIndex   : 13
InterfaceAlias    : Ethernet0
AddressFamily     : IPv6
Type             : Unicast
PrefixLength     : 128
PrefixOrigin     : RouterAdvertisement
SuffixOrigin     : Random
AddressState     : Preferred
ValidLifetime    : 6:23:48:23
PreferredLifetime : 23:41:46
SkipAsSource     : False
```



```

PolicyStore      : ActiveStore

IPAddress        : fd8f:cafe:1ab::d4d
InterfaceIndex   : 13
InterfaceAlias   : Ethernet0
AddressFamily    : IPv6
Type             : Unicast
PrefixLength     : 128
PrefixOrigin     : Dhcp
SuffixOrigin     : Dhcp
AddressState     : Preferred
ValidLifetime    : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource     : False
PolicyStore      : ActiveStore

IPAddress        : 2001:db8:f4:3bc:a973:10e1:e029:d1bb
InterfaceIndex   : 13
InterfaceAlias   : Ethernet0
AddressFamily    : IPv6
Type             : Unicast
PrefixLength     : 64
PrefixOrigin     : RouterAdvertisement
SuffixOrigin     : Link
AddressState     : Preferred
ValidLifetime    : 11.10:34:12
PreferredLifetime : 6.23:42:30
SkipAsSource     : False
PolicyStore      : ActiveStore

IPAddress        : 2001:db8:f4:3bc:7c80:3143:f2c8:8a05
InterfaceIndex   : 13
InterfaceAlias   : Ethernet0
AddressFamily    : IPv6
Type             : Unicast
PrefixLength     : 128
PrefixOrigin     : RouterAdvertisement
SuffixOrigin     : Random
AddressState     : Preferred
ValidLifetime    : 6.23:48:23
PreferredLifetime : 23:41:46
SkipAsSource     : False
PolicyStore      : ActiveStore

IPAddress        : 2001:db8:f4:3bc::d4d
InterfaceIndex   : 13
InterfaceAlias   : Ethernet0
AddressFamily    : IPv6
Type             : Unicast
PrefixLength     : 128
PrefixOrigin     : Dhcp
SuffixOrigin     : Dhcp
AddressState     : Preferred
ValidLifetime    : 11.10:34:12
PreferredLifetime : 6.23:42:30
SkipAsSource     : False
PolicyStore      : ActiveStore

```

```

IPAddress      : 2001:db8:f4:2b4::d4d
InterfaceIndex : 13
InterfaceAlias : Ethernet0
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 128
PrefixOrigin   : Dhcp
SuffixOrigin   : Dhcp
AddressState   : Preferred
ValidLifetime  : 13.05:44:04
PreferredLifetime : 6.05:44:04
SkipAsSource   : False
PolicyStore    : ActiveStore

IPAddress      : fe80::77:3985:3f57:fe54%7
InterfaceIndex : 7
InterfaceAlias : Teredo Tunneling Pseudo-Interface
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 64
PrefixOrigin   : WellKnown
SuffixOrigin   : Link
AddressState   : Preferred
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : ActiveStore

IPAddress      : 2001:0:9d38:6ab8:77:3985:3f57:fe54
InterfaceIndex : 7
InterfaceAlias : Teredo Tunneling Pseudo-Interface
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 64
PrefixOrigin   : RouterAdvertisement
SuffixOrigin   : Link
AddressState   : Preferred
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : ActiveStore

IPAddress      : ::1
InterfaceIndex : 1
InterfaceAlias : Loopback Pseudo-Interface 1
AddressFamily  : IPv6
Type           : Unicast
PrefixLength   : 128
PrefixOrigin   : WellKnown
SuffixOrigin   : WellKnown
AddressState   : Preferred
ValidLifetime  : Infinite ([TimeSpan]::MaxValue)
PreferredLifetime : Infinite ([TimeSpan]::MaxValue)
SkipAsSource   : False
PolicyStore    : ActiveStore

```

2.11. Get-NetNeighbor -AddressFamily IPv6

La commande `Get-NetNeighbor -AddressFamily IPv6` correspond à la commande `netsh interface ipv6 show neighbor`.

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\user> Get-NetNeighbor -AddressFamily IPv6

ifIndex	IPAddress	LinkLayerAddress	State	PolicyStore
-----	-----	-----	-----	-----
14	ff05::1:3	33-33-00-01-00-03	Permanent	ActiveStore
14	ff05::c	33-33-00-00-00-0C	Permanent	ActiveStore
14	ff02::1:ff0b:d525	33-33-FF-0B-D5-25	Permanent	ActiveStore
14	ff02::1:ff00:1	33-33-FF-00-00-01	Permanent	ActiveStore
14	ff02::1:3	33-33-00-01-00-03	Permanent	ActiveStore
14	ff02::1:2	33-33-00-01-00-02	Permanent	ActiveStore
14	ff02::16	33-33-00-00-00-16	Permanent	ActiveStore
14	ff02::c	33-33-00-00-00-0C	Permanent	ActiveStore
14	ff02::2	33-33-00-00-00-02	Permanent	ActiveStore
14	ff02::1	33-33-00-00-00-01	Permanent	ActiveStore
4	ff05::1:3	33-33-00-01-00-03	Permanent	ActiveStore
4	ff05::c	33-33-00-00-00-0C	Permanent	ActiveStore
4	ff02::1:ff00:1	33-33-FF-00-00-01	Permanent	ActiveStore
4	ff02::1:2	33-33-00-01-00-02	Permanent	ActiveStore
4	ff02::16	33-33-00-00-00-16	Permanent	ActiveStore
13	ff05::1:3	33-33-00-01-00-03	Permanent	ActiveStore
13	ff05::c	33-33-00-00-00-0C	Permanent	ActiveStore
13	ff02::1:fffe:2eee	33-33-FF-FE-2E-EE	Permanent	ActiveStore
13	ff02::1:ffe3:ecd8	33-33-FF-E3-EC-D8	Permanent	ActiveStore
13	ff02::1:ffc8:8a05	33-33-FF-C8-8A-05	Permanent	ActiveStore
13	ff02::1:ff29:d1bb	33-33-FF-29-D1-BB	Permanent	ActiveStore
13	ff02::1:ff00:d4d	33-33-FF-00-0D-4D	Permanent	ActiveStore

13	ff02::1:ff00:1	33-33-FF-00-00-01	Permanent	ActiveSto\
re				
13	ff02::1:3	33-33-00-01-00-03	Permanent	ActiveSto\
re				
13	ff02::1:2	33-33-00-01-00-02	Permanent	ActiveSto\
re				
13	ff02::fb	33-33-00-00-00-FB	Permanent	ActiveSto\
re				
13	ff02::16	33-33-00-00-00-16	Permanent	ActiveSto\
re				
13	ff02::c	33-33-00-00-00-0C	Permanent	ActiveSto\
re				
13	ff02::2	33-33-00-00-00-02	Permanent	ActiveSto\
re				
13	ff02::1	33-33-00-00-00-01	Permanent	ActiveSto\
re				
13	fe80::52c7:bfff:fefe:2eee	50-C7-BF-FE-2E-EE	Stale	ActiveSto\
re				
13	fd8f:cafe:1ab::1	50-C7-BF-FE-2E-EE	Stale	ActiveSto\
re				
7	ff05::1:3	255.255.255.255:655	Permanent	ActiveSto\
re				
7	ff05::c	255.255.255.255:655	Permanent	ActiveSto\
re				
7	ff02::1:ff00:1	255.255.255.255:655	Permanent	ActiveSto\
re				
7	ff02::1:2	255.255.255.255:655	Permanent	ActiveSto\
re				
7	ff02::16	255.255.255.255:655	Permanent	ActiveSto\
re				
7	ff02::2	255.255.255.255:655	Permanent	ActiveSto\
re				
7	fe80::8000:f227:62c7:9547	255.255.255.255:655	Unreachable	ActiveSto\
re				
7	2001:0:9d38:6ab8:ccd:209b:53eb:f5fc	255.255.255.255:655	Unreachable	ActiveSto\
re				
1	ff05::1:3		Permanent	ActiveSto\
re				
1	ff05::c		Permanent	ActiveSto\
re				
1	ff02::1:ff00:1		Permanent	ActiveSto\
re				
1	ff02::1:2		Permanent	ActiveSto\
re				
1	ff02::16		Permanent	ActiveSto\
re				
1	ff02::c		Permanent	ActiveSto\
re				
1	ff02::2		Permanent	ActiveSto\
re				

2.12.Get-NetRoute -AddressFamily IPv6

La commande `Get-NetRoute -AddressFamily IPv6` correspond à la commande `netsh interface ipv6 show route`.

```
PS C:\Users\user> Get-NetRoute -AddressFamily IPv6
```

ifIndex	DestinationPrefix	NextHop	RouteM\
etric ifMetric PolicyStore			
4	ff00::/8	::	\
256 65	ActiveStore		
7	ff00::/8	::	\
256 75	ActiveStore		
13	ff00::/8	::	\
256 25	ActiveStore		
14	ff00::/8	::	\
256 25	ActiveStore		
1	ff00::/8	::	\
256 75	ActiveStore		
14	fe80::bcfb:20de:f90b:d525/128	::	\
256 25	ActiveStore		
13	fe80::a973:10e1:e029:d1bb/128	::	\
256 25	ActiveStore		
4	fe80::68c0:6c2e:fe7c:4c9b/128	::	\
256 65	ActiveStore		
7	fe80::77:3985:3f57:fe54/128	::	\
256 75	ActiveStore		
7	fe80::/64	::	\
256 75	ActiveStore		
4	fe80::/64	::	\
256 65	ActiveStore		
13	fe80::/64	::	\
256 25	ActiveStore		
14	fe80::/64	::	\
256 25	ActiveStore		
13	fd8f:cafe:1ab:0:a973:10e1:e029:d1bb/128	::	\
256 25	ActiveStore		
13	fd8f:cafe:1ab:0:7c80:3143:f2c8:8a05/128	::	\
256 25	ActiveStore		
13	fd8f:cafe:1ab::d4d/128	::	\
256 25	ActiveStore		
13	fd8f:cafe:1ab::/64	::	\
256 25	ActiveStore		
13	fd8f:cafe:1ab::/48	fe80::52c7:bfff:fefe:2eee	\
256 25	ActiveStore		
13	2001:db8:f4:3bc:a973:10e1:e029:d1bb/128	::	\
256 25	ActiveStore		
13	2001:db8:f4:3bc:7c80:3143:f2c8:8a05/128	::	\
256 25	ActiveStore		
13	2001:db8:f4:3bc::d4d/128	::	\
256 25	ActiveStore		
13	2001:db8:f4:3bc::/64	fe80::52c7:bfff:fefe:2eee	\
256 25	ActiveStore		
13	2001:db8:f4:3bc::/64	::	\
256 25	ActiveStore		
13	2001:db8:f4:2b4::d4d/128	::	\
256 25	ActiveStore		
7	2001:0:9d38:6ab8:77:3985:3f57:fe54/128	::	\
256 75	ActiveStore		
7	2001::/32	::	\
256 75	ActiveStore		

```

1          ::1/128          ::          \
  256 75      ActiveStore
13         ::/0             fe80::52c7:bfff:fefe:2eee \
  256 25      ActiveStore

```

3. Hôte Linux

3.1. Liste des commandes

- `ip -6 add`
- `ip -6 neigh`
- `ip -6 maddress`
- `ip -6 route`

3.2. `ip -6 add`

La commande `ip -6 add` affiche les adresses IPv6 des interfaces.

```

[user@localhost ~]$ ip -6 add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 fd8f:cafe:1ab::3b0/128 scope global
        valid_lft forever preferred_lft forever
    inet6 2001:db8:f4:3bc::3b0/128 scope global dynamic
        valid_lft 988207sec preferred_lft 603505sec
    inet6 fd8f:cafe:1ab:0:defc:4132:c315:483/64 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 2001:db8:f4:3bc:aea7:d3e7:ea94:f9cc/64 scope global noprefixroute dynamic
        valid_lft 988207sec preferred_lft 603505sec
    inet6 fe80::9055:5984:36b8:c50c/64 scope link
        valid_lft forever preferred_lft forever

```

3.3. `ip -6 maddress`

La commande `ip -6 maddress` affiche les groupes Multicast joints.

```

1:      lo
    inet6 ff02::1
    inet6 ff01::1
2:      ens33
    inet6 ff02::1:ff00:3b0 users 2
    inet6 ff02::1:ff15:483
    inet6 ff02::1:ff94:f9cc
    inet6 ff02::1:ffb8:c50c
    inet6 ff02::1
    inet6 ff01::1
3:      virbr0
    inet6 ff02::1

```

```
        inet6 ff01::1
4:      virbr0-nic
        inet6 ff02::1
        inet6 ff01::1
```

3.4. ip -6 neigh

La commande `ip -6 neigh` affiche la table de voisinage IPv6.

```
[user@localhost ~]$ ip -6 neigh
fe80::52c7:bfff:fefe:2eee dev ens33 lladdr 50:c7:bf:fe:2e:ee router STALE
```

3.5. ip -6 route

La commande `ip -6 route` affiche la table de routage IPv6.

```
[user@localhost ~]$ ip -6 route
unreachable ::/96 dev lo metric 1024 error -113
unreachable ::ffff:0.0.0.0/96 dev lo metric 1024 error -113
unreachable 2002:a00::/24 dev lo metric 1024 error -113
unreachable 2002:7f00::/24 dev lo metric 1024 error -113
unreachable 2002:a9fe::/32 dev lo metric 1024 error -113
unreachable 2002:ac10::/28 dev lo metric 1024 error -113
unreachable 2002:c0a8::/32 dev lo metric 1024 error -113
unreachable 2002:e000::/19 dev lo metric 1024 error -113
2001:db8:f4:3bc::3b0 dev ens33 proto kernel metric 256 expires 988089sec
2001:db8:f4:3bc::/64 via fe80::52c7:bfff:fefe:2eee dev ens33 proto ra metric 100
unreachable 3ffe:ffff::/32 dev lo metric 1024 error -113
fd8f:cafe:1ab::3b0 dev ens33 proto kernel metric 256
fd8f:cafe:1ab::/64 dev ens33 proto ra metric 100
fd8f:cafe:1ab::/48 via fe80::52c7:bfff:fefe:2eee dev ens33 proto ra metric 100
fe80::52c7:bfff:fefe:2eee dev ens33 proto static metric 100
fe80::/64 dev ens33 proto kernel metric 256
default via fe80::52c7:bfff:fefe:2eee dev ens33 proto static metric 100
```

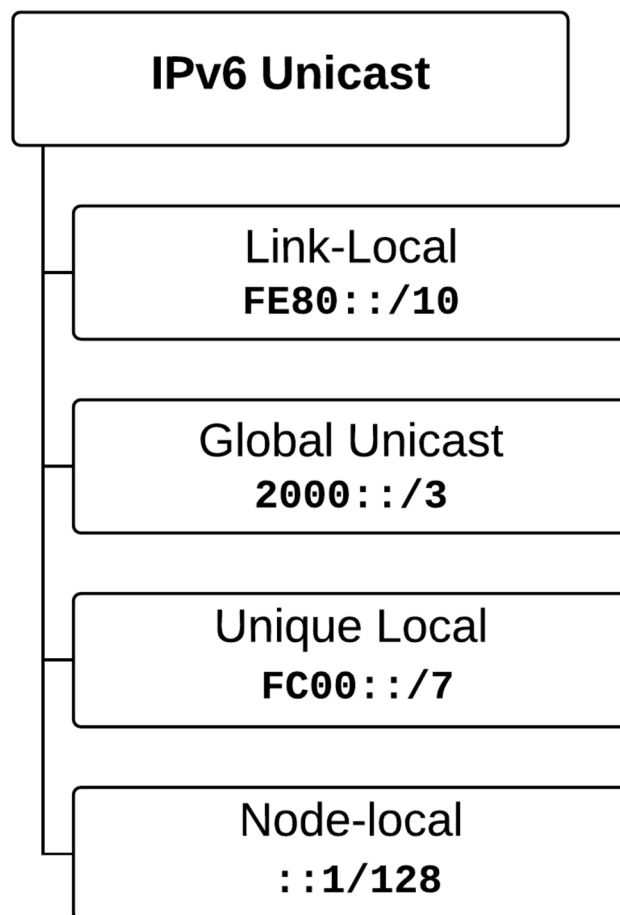
20. Les adresses IPv6 Unicast

Parmi les adresses IPv6 Unicast, on trouve plusieurs classes réservées à certains usages.

- **Node-Local** `::1/128` : adresse de bouclage
- **Link-Local** `fe80::/10` : adresse obligatoire et indispensable au bon fonctionnement du protocole
- **Global Unicast** `2000::/3` : adresse publique
- **Unique Local** `fc00::/7` (c'est le préfixe `fd00::/8` qui est utilisé : adresse privée (aléatoire))

Une adresse **Anycast** ne se distingue pas d'une adresse Unicast ; ce terme "Anycast" désigne une adresse de livraison qui atteindra sa destination par une technique de routage au plus proche ou au plus efficient.

On remarquera : Unspecified `::/128` : route non spécifiée, et Default Route `::/0` : route par défaut.



Adresses IPv6 Unicast

1. Adresse IPv6 Unicast de bouclage (`::1/128`)

On distinguera la notion d'*interface* de bouclage et de celle d'*adresse* de bouclage (*loopback*).

L'interface `lo` ne dispose pas d'adresse IPv6 Link-Local : elle ne peut d'ailleurs communiquer qu'avec elle-même. En IPv6, le bouclage indispensable aux systèmes d'exploitation est adressé par une seule adresse spéciale `::1/128`.

```
Microsoft Windows [Version 10.0.16299.309]
```

```
(c) 2017 Microsoft Corporation. All rights reserved.
```

```
C:\Users\user>netsh interface ipv6 show address
```

```
Interface 1: Loopback Pseudo-Interface 1
```

```
Addr Type  DAD State  Valid Life Pref. Life Address
```

```
-----
```

Addr	Type	DAD State	Valid Life	Pref. Life	Address
Other	Preferred		infinite	infinite	::1

```
(...)
```

2. Adressage IPv6 Link-Local Unicast

2.1. Adresses IPv6 Link-Local (`fe80::/10`)

Une [adresse Link-Local](#) est une adresse qui ne porte que sur le lien (L2) et qui n'est jamais transféré par les routeurs. Elle sert à joindre les voisins sur un même lien. Toutes les interfaces IPv6 disposent d'une adresse Link-Local.

Ses 10 premiers bits sont codés en `1111111010` pour donner en hexadécimal un bloc `fe80::/10`.

1111111010 (10)	(54)	Interface ID (64)
---------------------------	-------------	--------------------------

Adresses IPv6 Link-Local

2.2. Caractéristiques d'une adresse IPv6 Link-Local

- Une adresse Link-Local est **obligatoire** sur chaque interface activée en IPv6.
- Elle est générée **automatiquement** et elle est censée être **unique** sur le lien.
- On la reconnaît par son préfixe `fe80::/10` ou ses dix premiers bits à `1111111010`.
- Cette destination est **purement locale** sur la liaison de l'interface. Les routeurs IPv6 **ne transfèrent pas** cette destination.
- Leurs durées de vie *Valid Lifetime* et *Preferred Lifetime* sont infinies.
- Configurées sur les interfaces leur **masque** est obligatoirement `/64`.

2.3. Reconnaître une adresse IPv6 Link-local

Dans cette sortie, on reconnaît quatre interfaces, chacune avec une adresse IPv6 Link-Local, hors l'interface de `loopback`.

- Interface 13: Teredo Tunneling Pseudo-Interface : `fe80::1010:1e8f :3f57:fe54%13`
- Interface 12: Ethernet0 : `fe80::a973:10e1:e029:d1bb%12`
- Interface 4: Bluetooth Network Connection : `fe80::68c0:6c2e :fe7c :4c9b%4`

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.
```

```
C:\Users\user>netsh interface ipv6 show address
```

```
Interface 1: Loopback Pseudo-Interface 1
```

Addr Type	DAD State	Valid Life	Pref. Life	Address
Other	Preferred	infinite	infinite	::1

```
Interface 13: Teredo Tunneling Pseudo-Interface
```

Addr Type	DAD State	Valid Life	Pref. Life	Address
Public	Preferred	infinite	infinite	2001:0:9d38:6abd:1010:1e8f:3f57:fe54
Other	Preferred	infinite	infinite	fe80::1010:1e8f:3f57:fe54%13

```
Interface 12: Ethernet0
```

Addr Type	DAD State	Valid Life	Pref. Life	Address
Dhcp	Preferred	10d12h15m6s	3d22h11m28s	2001:db8:f4:4b3::d4d
Public	Preferred	10d12h15m7s	3d22h11m29s	2001:db8:f4:4b3:a973:10e1:e029:d1bb
Temporary	Preferred	6d23h55m39s	23h47m26s	2001:db8:f4:4b3:b937:ea7c:7175:5743
Dhcp	Preferred	10d12h15m6s	3d22h11m28s	2001:db8:f4:1ded::d4d
Public	Preferred	10d12h15m7s	3d22h11m29s	2001:db8:f4:1ded:a973:10e1:e029:d1bb
Temporary	Preferred	6d23h55m39s	23h47m26s	2001:db8:f4:1ded:b937:ea7c:7175:5743
Public	Preferred	infinite	infinite	fd8f:cafe:1ab:0:a973:10e1:e029:d1bb
Temporary	Preferred	6d23h55m39s	23h47m26s	fd8f:cafe:1ab:0:b937:ea7c:7175:5743
Other	Preferred	infinite	infinite	fe80::a973:10e1:e029:d1bb%12

```
Interface 4: Bluetooth Network Connection
```

Addr Type	DAD State	Valid Life	Pref. Life	Address
Other	Deprecated	infinite	infinite	fe80::68c0:6c2e:fe7c:4c9b%4

2.4. Utilité d'une adresse IPv6 Link-Local

Quel est leur usage ? On peut les comparer *légèrement* aux adresses APIPA IPv4 169.254.0.0/16.

Mais Neighbor Discovery (ND) indispensable au fonctionnement d'IPv6, DHCPv6 (à des fins de gestion), les protocoles de routage utilisent aussi ces adresses. Leur usage est devenu un standard. On peut aussi les utiliser pour héberger un service local ou pour faire du diagnostic ICMPv6.

Attention, une interface ne peut pas fonctionner sans disposer d'une telle adresse Link-Local ! Elle est indispensable au fonctionnement ultérieur d'IPv6 sur la connexion : découverte de routes, de routeurs, de serveurs de nom ou DHCPv6. Chaque interface activée en IPv6 doit disposer de ce type d'interface.

2.5. Portée d'une adresse IPv6 Link-Local

Aussi, on remarquera le modulo % suivi du chiffre X. Cette valeur identifie dans Windows l'interface associée à l'adresse. En effet, sur un ordinateur ou sur un routeur disposant de plusieurs interfaces, on trouvera le même bloc IPv6 fe80::/64 utilisé par plusieurs interfaces qui ne transfèrent pas ce trafic. Elles sont nécessairement associées par les systèmes avec leur interface de sortie.

3. Adressage IPv6 Global Unicast (2000::/3)

3.1. Adresses IPv6 Global Unicast

On pourrait identifier plusieurs adresses Global Unicast sur les interfaces, soit l'équivalent de nos adresses IPv4 publiques. On les définira plus précisément comme des destinations publiées sur l'Internet. Les routeurs transfèrent le trafic vers ces destinations. Elles sont donc "globalement routables".

3.2. Format des adresses Global Unicast

Le format des adresses IPv6 Global Unicast est défini dans les [RFC 3587](#) et [RFC 3177](#) :

001 (3)	Global Routing Prefix (45)	Subnet ID (16)	Interface ID (64)
------------	-------------------------------	-------------------	-------------------

Adresses IPv6 Global Unicast

- Le "global routing prefix" est la partie de l'adresse assignée à un site (c'est-à-dire un ensemble de sous-réseaux et de liens). Ces numéros sont attribués par les RIRs et les ISPs. Il a une longueur par défaut de 48 bits.
- Le "subnet ID" est l'identifiant du sous-réseau dans le site. Il est géré par les administrateurs du site. Il a une longueur de 16 bits portant le masque de sous-réseau à 64 bits.
- L'"Interface ID" identifie l'interface dans le sous-réseau. C'est élément de 64 bits qui est configuré de différentes manières.

3.3. Caractéristiques des adresses IPv6 Global Unicast

Le bloc 2000::/3 est réservé à cet usage. On identifie ces adresses par une valeur comprise entre 0x2000 à 0x3FFF sur les 4 premiers hexas de l'adresse.

Avec un tel masque, c'est 1/8 de l'espace IPv6 qui a été réservé à un usage public. En réalité la seconde moitié du bloc (3000::/4) est réservée par l'IANA de telle sorte que seul le bloc 2000::/4 soit utilisé. Concrètement, toute adresse commençant par son premier hexa à 2 est une adresse global Unicast.

Les quatre premiers bits d'une adresse IPv6 Global unicast sont toujours 0001. De manière plus "legacy", on peut affirmer que les trois derniers bits du premier "mot" d'une adresse IPv6 Global Unicast sont tout à zéro (000).

3.4. Attribution des adresses IPv6 Global Unicast

[IPv6 Global Unicast Address Assignments](#)

Blocs réservés dans 2000::/3

- 2d00:0000::/8 IANA 1999-07-01 RESERVED
- 2e00:0000::/7 IANA 1999-07-01 RESERVED
- 2001:0000::/23 IANA 1999-07-01 whois.iana.or ALLOCATED
 - 2001:0000::/23 is reserved for IETF Protocol Assignments [RFC2928](#).
 - 2001:0000::/32 is reserved for TEREDO [RFC4380](#).
 - 2001:1::1/128 is reserved for Port Control Protocol Anycast [RFC7723](#).
 - 2001:2::/48 is reserved for Benchmarking [RFC5180](#) [RFC Errata 1752].
 - 2001:3::/32 is reserved for AMT [RFC7450](#).
 - 2001:4:112::/48 is reserved for AS112-v6 [RFC7535](#).

- 2001:5::/32 is reserved for EID Space for LISP [RFC7954](#).
- 2001:10::/28 is deprecated (previously ORCHID) [RFC4843](#).
- 2001:20::/28 is reserved for ORCHIDv2 [RFC7343](#).
- 2001:db8::/32 is reserved for Documentation [RFC3849](#).
- 2002:0000::/16 is reserved for 6to4 [RFC3056](#).
- 3000:0000::/4 IANA 1999-07-01 RESERVED
- 3ffe::/16 IANA 2008-04 RESERVED
 - 3ffe:831f::/32 was used for Teredo in some old but widely distributed networking stacks. This usage is deprecated in favor of 2001::/32, which was allocated for the purpose in [RFC4380](#). 3ffe::/16 and 5f00::/8 were used for the 6bone but were returned. [RFC5156]

Blocs routables par RIRs



RIRs (Regional Internet Registries)

Source de l'image.

AFRINIC

- 2001:4200::/23
- 2c00:0000::/12

APNIC

- 2001:0200::/23
- 2001:0e00::/23
- 2001:0c00::/23
- 2001:4400::/23
- 2001:8000::/19

- 2001:a000::/20
- 2001:b000::/20
- 2400:0000::/12

ARIN

- 2001:0400::/23
- 2001:1800::/23
- 2001:4800::/23
- 2600:0000::/12
- 2610:0000::/23
- 2620:0000::/23

LACNIC

- 2001:1200::/23
- 2800:0000::/12

RIPE-NCC

- 2001:0600::/23
- 2001:0800::/23
- 2001:0a00::/23
- 2001:1400::/23
- 2001:1600::/23
- 2001:1a00::/23
- 2001:1c00::/22
- 2001:2000::/20
- 2001:3000::/21
- 2001:3800::/22
- 2001:3c00::/22
- 2001:4000::/23
- 2001:4600::/23
- 2001:4a00::/23
- 2001:4c00::/23
- 2001:5000::/20

- 2003:0000::/18
- 2a00:0000::/12

3.5. Comment reconnaître les adresses IPv6 Global Unicast ?

Voici une sortie focalisée sur l'interface Ethernet0 d'un ordinateur Windows :

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\user>netsh interface ipv6 show address

(...)

Interface 12: Ethernet0

Addr Type   DAD State   Valid Life Pref. Life Address
-----
Dhcp        Preferred  10d12h15m6s 3d22h11m28s 2001:db8:f4:4b3::d4d
Public      Preferred  10d12h15m7s 3d22h11m29s 2001:db8:f4:4b3:a973:10e1:e029:d1bb
Temporary   Preferred  6d23h55m39s 23h47m26s 2001:db8:f4:4b3:b937:ea7c:7175:5743
Dhcp        Preferred  10d12h15m6s 3d22h11m28s 2001:db8:f4:1ded::d4d
Public      Preferred  10d12h15m7s 3d22h11m29s 2001:db8:f4:1ded:a973:10e1:e029:d1bb
Temporary   Preferred  6d23h55m39s 23h47m26s 2001:db8:f4:1ded:b937:ea7c:7175:5743
Public      Preferred  infinite    infinite    fd8f:cafe:1ab:0:a973:10e1:e029:d1bb
Temporary   Preferred  6d23h55m39s 23h47m26s  fd8f:cafe:1ab:0:b937:ea7c:7175:5743
Other        Preferred  infinite    infinite    fe80::a973:10e1:e029:d1bb%12

(...)
```

On remarquera ici trois adresses IPv6 Global Unicast dans le bloc 2001:db8:f4:4b3::/64 :

```
2001:db8:f4:4b3::d4d
2001:db8:f4:4b3:a973:10e1:e029:d1bb
2001:db8:f4:4b3:b937:ea7c:7175:5743
```

et trois autres dans le bloc :

```
2001:db8:f4:1ded::d4d
2001:db8:f4:1ded:a973:10e1:e029:d1bb
2001:db8:f4:1ded:b937:ea7c:7175:5743
```

On reconnaîtra le préfixe de documentation 2001:db8::/32 qui a été adapté pour des raisons de vie privée de la connexion réelle qui sert à l'illustration.

Pour chaque bloc, la première adresse a été attribuée de manière arbitraire par un service DHCPv6. Les deux suivantes sont des adresses dont l'identifiant d'interface a été généré aléatoirement.

Si l'on s'attarde sur les deux dernières adresses générées manifestement de manière aléatoire, elles disposent de durées de vie limitées. L'une, la "publique", sert de destination venant de l'Internet (activez et mettez à jour vos pare-feu IPv6); l'autre, l'adresse temporaire sert toujours comme adresse d'origine du trafic de l'interface pour le trafic à destination de l'Internet. Dans ce cas, l'adresse attribuée par DHCPv6 ne peut servir que comme adresse de destination. Je l'ai implémentée ici à des fins de gestion (résolution dynamique de noms).

4. Adressage IPv6 Unique Local (FC00::/7)

4.1. Adresses IPv6 Unique Local

L'adressage IPv6 Unique Local est défini dans le [RFC 4193, Unique Local IPv6 Unicast Address](#).

On identifiera un autre type d'adresses : Unique Local Address (ULA). Comparables aux adresses privées IPv4 RFC1918 10.0.0.0/8, 172.16.0.0/12 et 192.168.0.0/16 dans le sens où elles **ne sont pas** "globalement routables", elles ne connaissent pas de destination sur l'Internet. On les reconnaît par leur préfixe FD00::/8 dont les 40 bits suivants ont été générés aléatoirement pour compléter un préfixe /48.

Cet adressage privé IPv6 est censé être unique afin d'éviter un chevauchement d'adresses identiques à chaque extrémité d'une connexion VPN par exemple.

4.2. Caractéristiques des adresses IPv6 Unique Local

- Elles ne connaissent pas de destination dans l'Internet public.
- Elle utilisent un préfixe qui est globalement unique (on dira avec une haute probabilité d'unicité).
- Ce préfixe est bien connu : FC00::/7 mais FD00::/8 notamment afin de faciliter le filtrage aux limites des sites.
- Elle permettent aux sites privés de s'interconnecter entre eux en réduisant la probabilité de conflits sur des blocs d'adresses non uniques qui se chevauchent.
- En cas de fuite DNS, il y a peu de risque de conflit avec d'autres adresses.

4.3. Utilité des adresses IPv6 Unique Local

Quel pourrait être leur usage au XXIème siècle avec un adressage global IPv6 aussi large ?

D'abord, elles sont recommandées par le [RFC7084 "Basic Requirements for IPv6 Customer Edge Routers"](#).

On peut les utiliser pour identifier des destinations privées entre des sites distants entre Paris et Lille, entre Bruxelles et Londres ou entre le bâtiment de la rue du commerce et celui du Boulevard du Nord ... à travers des connexions dédiées ou Internet sécurisées par IPSEC En deux mots, faciliter le routage privé.

4.4. Format des adresses IPv6 Unique Local

Les adresses IPv6 Unique Local sont créées sur base d'un identifiant global généré de manière pseudo-aléatoire en fonction du temps (NTP) et de l'identifiant unique EUI-64 concaténés pour créer une clé SHA-1 et générer une valeur de 160 bits dont les 40 derniers bits significatifs remplissent le champs "Global ID".

1111110 (7)	1 (L)	Global ID (40)	Subnet ID (16)	Interface ID (64)
----------------	----------	-------------------	-------------------	-------------------

Adresses IPv6 Unique Local

Où :

- Prefix : FC00::/7 ou 1111110 sur les 7 premiers bits identifie le préfixe des adresses IPv6 Unique Local.
- L : Valeur à 1 pour un préfixe "localement" assigné (0 est réservé à un usage futur, jamais ?). Le préfixe devient *de facto* FD00::/8 où les 8 premiers bits de l'adresse sont 11111101
- Global ID : 40-bit "Global ID" identifiant global généré de manière pseudo-aléatoire pour créer un préfixe unique de site. Il complète un /48 de préfixe de site.

- Subnet ID : 16-bit “Subnet ID” comme identifiant du sous-réseau dans le site.
- Interface ID : 64-bit “Interface ID” qui identifie l’interface dans le sous-réseau.

4.5. Comment reconnaître des adresses IPv6 Unique Local ?

Voici une sortie focalisée sur l’interface Ethernet0 d’un ordinateur Windows :

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\user>netsh interface ipv6 show address

(...)

Interface 12: Ethernet0

Addr Type   DAD State   Valid Life Pref. Life Address
-----
Dhcp        Preferred  10d12h15m6s 3d22h11m28s 2001:db8:f4:4b3::d4d
Public      Preferred  10d12h15m7s 3d22h11m29s 2001:db8:f4:4b3:a973:10e1:e029:d1bb
Temporary   Preferred  6d23h55m39s 23h47m26s 2001:db8:f4:4b3:b937:ea7c:7175:5743
Dhcp        Preferred  10d12h15m6s 3d22h11m28s 2001:db8:f4:1ded::d4d
Public      Preferred  10d12h15m7s 3d22h11m29s 2001:db8:f4:1ded:a973:10e1:e029:d1bb
Temporary   Preferred  6d23h55m39s 23h47m26s 2001:db8:f4:1ded:b937:ea7c:7175:5743
Public      Preferred  infinite    infinite  fd8f:cafe:1ab:0:a973:10e1:e029:d1bb
Temporary   Preferred  6d23h55m39s 23h47m26s  fd8f:cafe:1ab:0:b937:ea7c:7175:5743
Other       Preferred  infinite    infinite  fe80::a973:10e1:e029:d1bb%12

(...)
```

Dans notre exemple, on trouve deux adresses IPv6 Unique Local autoconfigurées.

```
fd8f:cafe:1ab:0:a973:10e1:e029:d1bb
fd8f:cafe:1ab:0:b937:ea7c:7175:5743
```

On constate que le préfixe attribué est `fd8f :cafe :1ab ::/64` qui, s’il n’a manifestement pas été généré de manière aléatoire selon le [RFC 4193](#) car il a été choisi, est supposé “globalement unique”.

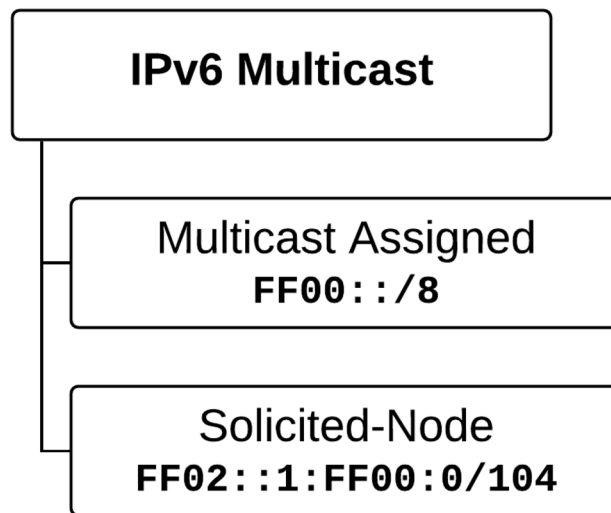
4.6. Développement du RFC 4193

Le [RFC 4193](#) développe plusieurs considérations sur les adresse Unique Local

- Comment les allouer.
- Considérations sur leur usage :
 - dans le routage
 - sur les routeurs de bordure de site,
 - en DNS,
 - dans le support des applications,
 - dans l’usage de VPN,
- des lignes directrices pour une communication au sein d’un site.

21. Les adresses IPv6 Multicast

Les différents types d'adresse IPv6 Multicast *Well-Know Multicast* et *Sollicted-Node Multicast* sont identifiées et comparées dans ce chapitre.



Types d'adresse IPv6 Multicast

1. Introduction aux adresses IPv6 Multicast FF00::/8

La seconde catégorie d'adresses IPv6 est celle des adresses Multicast, à destination de certaines interfaces sensibles à une adresse dans laquelle elles se reconnaissent. Pour ce faire, on dit qu'une interface est inscrite dans un groupe Multicast. Une adresse Multicast est une seule adresse de destination à laquelle répondent une ou plusieurs interfaces (plusieurs noeuds).

Une des intentions du Multicast est de remplacer le Broadcast (disparu en IPv6) qui identifie toutes les interfaces des hôtes du réseau. Le trafic Multicast est traduit par une adresse MAC avec un préfixe de 24 bits 33:33:00 et sera traité par les commutateurs comme du trafic de Broadcast. Une optimisation consisterait à déployer des protocoles qui permettent d'identifier les ports inscrits dans des groupes Multicast. Toutefois, il semblerait que le nombre de groupes dans lequel pourrait s'inscrire chaque interface IPv6 ne permettrait pas aux commutateurs actuels de tenir la charge.

2. Types d'adresse Multicast

On trouvera deux types d'adresses Multicast :

- Les adresses IPv6 **Well-Know Multicast**, connues d'avance, qui identifient des services bien connus,
- Les adresses IPv6 **Sollicted-Node Multicast**, qui servent à joindre une interface dont on veut connaître l'adresse MAC (Neighbor Solicitation).

3. Format d'une adresse IPv6 Multicast

Le format d'une adresse Multicast est indique dans le [RFC 4291](#).

11111111 (8)	flgs (4)	scop (4)	group ID (112)
------------------------	--------------------	--------------------	-----------------------

Format d'une adresse IPv6 Multicast

Nécessairement les huit premiers bits d'une adresse Multicast sont "tout-à-un" et donnent en hexadécimal une adresse qui commence par FF.

On identifiera le champ scop qui indique la portée du groupe :

- 1- Interface-Local scope
- 2- Link-Local scope
- 4- Admin-Local scope
- 5- Site-Local scope
- 8- Organization-Local scope
- E- Global scope

Sur les seconds huit bits de l'adresse on trouve tous les champs flgs à zéro et le champ scop à deux : 00000010, de telle sorte que les adresses Multicast ont la plupart du temps un premier mot à FF02. Le FF02 indique donc une portée uniquement locale, sur la facilité L2, sur le LAN.

4. Identifiant de groupe

- ff01::1 : tous les noeuds en local,
- ff02::1 : tous les noeuds sur le réseau local,
- ff02::2 : tous les routeurs sur le réseau local,
- ff02::5 : tous les routeurs OSPF
- ff02::6 : tous les routeurs OSPF DR/BDR
- ff02::9 : tous les routeurs RIP
- ff02::a : tous les routeurs EIGRP
- ff02::1:2 : tous les serveurs DHCP sur le réseau local
- ff02::fb : DNS Multicast sur le réseau local
- ff02::101 : Tous les serveurs NTP sur le réseau local
- ff02::c : SSDP (UPnP sous Windows)
- ff02::1:3 : Link-local Multicast Name Resolution

5. Identifier les groupes Multicast d'une interface IPv6

La commande "netsh interface ipv6 show joins" nous présente les groupes Multicast auxquels l'interface ethernet0 a souscrit :

```
Microsoft Windows [Version 10.0.16299.309]
(c) 2017 Microsoft Corporation. All rights reserved.
```

```
C:\Users\user>netsh interface ipv6 show joins
```

```
...
```

```
Interface 12: Ethernet0
```

Scope	References	Last	Address
0	0	Yes	ff01::1
0	0	Yes	ff02::1
0	1	Yes	ff02::c
0	1	Yes	ff02::fb
0	1	Yes	ff02::1:3
0	2	Yes	ff02::1:ff00:d4d
0	4	Yes	ff02::1:ff29:d1bb
0	3	Yes	ff02::1:ff75:5743

On reconnaît les adresses Multicast par leur préfixe FF00::/8.

6. Adresse IPv6 Solicited-Node Multicast FF02::1:ff00:0/104

Une adresse IPv6 Solicited-Node Multicast sert à joindre une interface dont on veut connaître l'adresse de couche 2.

En IPv4, les tables ARP sont remplies grâce à des messages ARP Request directement adressés en adresse MAC de Broadcast FF :FF :FF :FF :FF :FF.

En IPv6 c'est Neighbor Discovery (ND) avec des messages ICMPv6 Neighbor Solicitation/Neighbor Advertisements qui remplit cette fonction. Étant donné que le Broadcast n'existe plus en IPv6, quelle est l'adresse à utiliser pour désigner en IPv6 et en adresse de couche L2 l'hôte de destination puisqu'il s'agit justement de prendre connaissance de l'adresse L2 ?

Une adresse IPv6 **Solicited-Node Multicast** est composée de 104 bits de préfixe FF02::1:ff00:0/104 complétés par les 24 derniers bits de l'adresse IPv6 à joindre.

Dans l'exemple suivant, toutes ces adresses ont leur point commun : leur 24 derniers bits.

Si l'adresse à joindre est 2001:db8:f4:4b3::d4d ou 2001:db8:f4:1ded::d4d alors l'adresse Multicast correspondante est ff02::1:ff00:d4d.

Mais également si l'adresse à joindre est 2001:db8:f4:4b3:a973:10e1:e029:d1bb ou 2001:db8:f4:1ded:a973:10e1:e029:d1bb ou encore fd8f:cafe:1ab:0:a973:10e1:e029:d1bb alors l'adresse Multicast correspondante est ff02::1:ff29:d1bb.

Les interfaces configurées avec les adresses 2001:db8:f4:4b3:b937:ea7c:7175:5743, 2001:db8:f4:1ded:b937:ea7c:7175:5743, fd8f:cafe:1ab:0:b937:ea7c:7175:5743 ont une adresse Solicited Node Multicast ff02::1:ff75:5743.

Dans notre exemple, les trois adresses IPv6 Solicited-Node Multicast correspondent :

- à l'adresse lien local
- aux adresses attribuées par DHCPv6
- aux adresses autoconfigurées publiques
- aux adresses autoconfigurées temporaire

et signifient que l'interface répondra à toutes ces adresses !

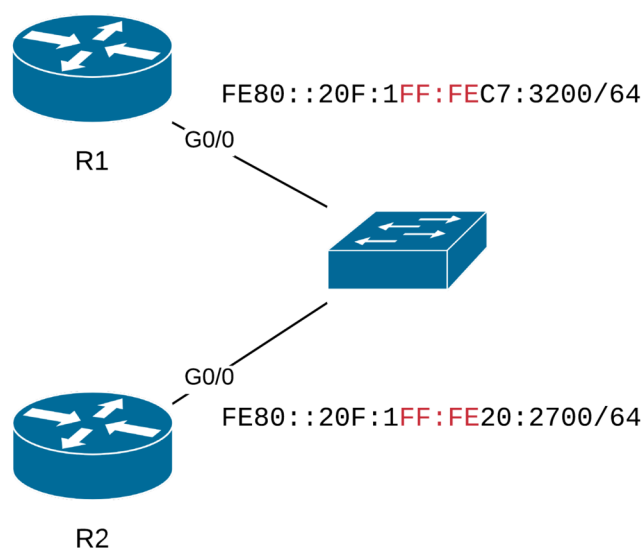
22. Configuration des interfaces IPv6 en Cisco IOS

Comment activer IPv6 sous Cisco IOS ? Comment vérifier la connectivité locale IPv6 sous Cisco IOS ? Comment consulter la table de voisinage ? Comment configurer une adresse Link-Local statique ? Et bien d'autres questions sur IPv6 sous Cisco IOS trouvent leur réponse ici.

1. Comment activer IPv6 sous Cisco IOS ?

Par défaut, les interfaces de routeurs Cisco **ne sont pas activées en IPv6**

Dans cette topologie de base nous considérons les routeurs Cisco comme des noeuds terminaux.



Topologie configuration des interfaces IPv6 sous Cisco IOS

A condition que l'interface G0/0 soit administrativement activée :

```
R1#configure terminal
R1(config)#interface G0/0
R1(config-if)#no shutdown
R1(config-if)#exit
R1(config)#exit
```

On constate bien qu'IPv6 est désactivé sur l'interface :

```
R1#show ipv6 interface G0/0
```

En effet, la commande `show ipv6 interface G0/0` ne donne aucun résultat.

On active IPv6 sur une interface avec la commande `ipv6 enable`. On notera que la configuration d'une adresse IPv6 statique rend cette étape accessoire en Cisco IOS.

```
R1#configure terminal
R1(config)#interface G0/0
R1(config-if)#ipv6 enable
R1(config-if)#exit
R1(config)#exit
```

On peut maintenant vérifier les paramètres IPv6 de l'interface G0/0 :

```
R1#show ipv6 interface
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::20F:1FF:FEC7:3200
  No Virtual link-local address(es):
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::1:FFC7:3200
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachable are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND NS retransmit interval is 1000 milliseconds
```

On apprend que cette interface GigabitEthernet0/0 répondra à trois adresses :

- Une **adresse Link-Local auto-configurée** (Stateless Address Auto Configuration) FE80::20F :1FF :FEC7:3200
- Une **adresse Well-Know Multicast** “Tous les noeuds sur la liaison” FF02::1
- Une **adresse Solicited-Node Multicast** FF02::1:FFC7:3200

On reconnaîtra dans les 24 derniers bits de l'adresse “Solicited-Node Multicast” ceux de l'adresse MAC 0xc73200.

2. Comment vérifier la connectivité locale IPv6 sous Cisco IOS ?

Sans garantie de succès (notamment à cause des pare-feu), on peut tenter de joindre “tous les noeuds sur la liaison locale” grâce à l'adresse FF02::1 avec la commande ping (en ICMPv6) :

```
R1#ping FF02::1
Output Interface: GigabitEthernet0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FF02::1, timeout is 2 seconds:
Packet sent with a source address of FE80::20F:1FF:FEC7:3200%GigabitEthernet0/0

Reply to request 0 received from FE80::20F:1FF:FE20:2700, 15 ms
Reply to request 1 received from FE80::20F:1FF:FE20:2700, 3 ms
Reply to request 2 received from FE80::20F:1FF:FE20:2700, 2 ms
Reply to request 3 received from FE80::20F:1FF:FE20:2700, 2 ms
Reply to request 4 received from FE80::20F:1FF:FE20:2700, 2 ms
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/4/15 ms
5 multicast replies and 0 errors.
```

Quand on veut joindre une destination Link-Local, il est nécessaire de préciser l'interface de sortie, ici GigabitEthernet0/0.

On constate que l'adresse d'un hôte IPv6 répond FE80::20F :1FF :FE20:2700 au premier paquet ICMPv6 envoyé avec la commande ping. Cette adresse IPv6 est-elle joignable ?

```

R1#ping FE80::20F:1FF:FE20:2700
Output Interface: GigabitEthernet0/0
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::20F:1FF:FE20:2700, timeout is 2 seconds:
Packet sent with a source address of FE80::20F:1FF:FEC7:3200%GigabitEthernet0/0
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4 ms

```

3. Comment consulter la table de voisinage ?

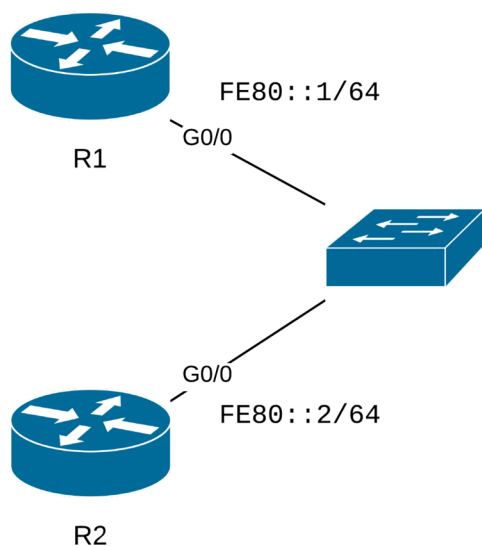
Ne cherchez plus la table ARP, en IPv6 cela s'appelle la "table de voisinage" :

```

R1#show ipv6 neighbors
IPv6 Address                               Age Link-layer Addr State Interface
FE80::20F:1FF:FE20:2700                    24 000f.0120.2700 STALE Gi0/0

```

4. Comment configurer une adresse Link-Local statique ?



Topologie configuration une adresse Link-Local statique

La commande `ipv6 address FE80::1 link-local` en configuration d'interface fixe l'adresse FE80::1.

```

R1#configure terminal
R1(config)#interface G0/0
R1(config-if)#ipv6 address FE80::1 link-local
R1(config-if)#exit
R1(config)#exit

```

Après vérification, l'interface sera à l'écoute de l'adresse FE80::1 mais aussi des adresses Multicast FF02::1 et FF02::1:FF00:1.

```
show ipv6 interface
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::1
  No Virtual link-local address(es):
  No global unicast address is configured
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachables are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND NS retransmit interval is 1000 milliseconds
```

5. Questions bonus sur IPv6

Questions pratiques

- Comment joindre tous les routeurs en IPv6 ?
- Est-ce que l'on peut configurer la même adresse "Link-Local" sur des interfaces différentes d'un routeur ?
- Qu'est-ce qu'une adresse IPv6 "Solicited-Node Multicast" ? Quelle est la commande qui permet de vérifier les groupes multicast d'une interface Cisco ?
- Que signifie le modulo % dans l'adresse IPv6 "Link-Local" de ma connexion ?
- Mon PC dispose de plusieurs adresses IPv6 publiques. Laquelle est utilisée pour accéder à Internet ? Suis-je vulnérable ?
- Que peut signifier le symbole :: en IPv6 ?

Questions théoriques

- Comment fonctionne "Stateless Address Auto Configuration" ?
- Quelle est l'étendue de l'adressage public IPv6 ?
- A quoi sert l'adressage privé (ULA) IPv6 ?

23. Plans d'adressage IPv6

Ce document a pour ambition de montrer la finesse et le confort que peut offrir un plan d'adressage IPv6.

1. Introduction aux plans d'adresses IPv6

Les objectifs de ce document sur les plans d'adressage IPv6 sont les suivants :

- Manipuler les bits, octets, mots en hexa
- Concevoir des plans d'adressage simples
- Concevoir des plans d'adressage hiérarchiques et évolutifs
- Adresser une topologie de laboratoire

2. Avantages d'un plan d'adressage

- Les tables de routage peuvent être réduites et être plus efficaces dans le processus de décision.
- Les politiques de sécurité peuvent être déployées plus facilement.
- Des politiques basées sur les applications peuvent être déployées.
- La gestion et l'approvisionnement du réseau peuvent être facilités.
- Le diagnostic est facilité, notamment par une meilleure identification.
- Mise à l'échelle facilitée suite à l'ajout de périphériques ou de sites.

3. Buts d'un plan d'adressage

- Fournir une capacité d'approvisionnement : connecter un nombre indéterminé de périphériques.
- Activer (ou non) les capacités de communication des hôtes : à communiquer entre eux (BYOD), communication interréseau (VLANs).
- Activer (ou non) la communication Internet : on peut orienter/filtrer le routage de certains préfixes ou de certains hôtes.
- Activer ou non le support d'applications : le filtrage par préfixe permettrait de faciliter le traitement de trafic multimédia ou vocal par exemple (VLANs).
- Permettre de mieux identifier les hôtes par niveau, emplacement géographique ou organisationnel, par fonction.

4. Attribution des blocs d'adresses IPv6

1. Il n'est plus recommandé aux fournisseurs d'accès Internet d'assigner des /128.
2. On a recommandé avec le RFC 3177 de blocs /48, /64 et /128.
3. Mais le [RFC 6177](#) ne recommande plus que des /48 au minimum.

5. Adresses IPv6 Provider-Aggregatable (PA) et Provider-Independent (PI)

Soit, un Fournisseur d'Accès Internet (LIR) attribue directement des blocs en "adresses IPv6 PA" pour "Provider-Aggregatable" au client final.

Il est possible d'obtenir un bloc routé indépendamment de son fournisseur d'accès [sur demande](#) : des adresses "IPv6 PI" pour "Provider-Independent" ; elles sont attribuées au client final directement par le RIR.

6. Principe de découpage

Le donné est toujours le même. On dispose d'un bloc d'adresses IP. On le découpe en sous-réseaux.

En IPv6 le masque le plus restrictif est un /64.

À la maison, chaque connexion domestique devrait disposer d'une connectivité publique avec un /64. Il est probable que vous ayez à gérer des blocs /48 attribués à votre connexion d'entreprise.

Avec un bloc /48, il reste un mot de 16 bits à découper en 1, en 16, en 256 ou en 4096 sous réseaux ...

On travaille alors principalement sur le quatrième mot d'une adresse.

7. Plan d'adressage Simple

On vous livre un bloc fdd4:478f :0611::/48

On peut "conformer" les adresses IPv4 privées ou adresses IPv6, par exemple :

- 172.16.1.0/24 $\Rightarrow$ fdd4:478f :0611:1::/64,
- 172.16.255.0/24 $\Rightarrow$ fdd4:478f :0611:255::/64,
- ...

On peut utiliser un découpage plat, à la volée :

- fdd4:478f :0611:0::/64,
- fdd4:478f :0611:1::/64,
- fdd4:478f :0611:2::/64,
- fdd4:478f :0611:3::/64, ...

8. Plan d'adressage à 2 ou 4 niveaux égaux

On vous livre un bloc fdd4:478f :0611::/48

Il reste 16 bits de libres, soit 4 hexas pour plusieurs stratégies hiérarchiques avec des niveaux géographiques, organisationnels, fonctionnels :

- Stratégie A : en deux niveaux égaux :
 - Niveau 1 : 256 réseaux (8 bits)/56
 - Niveau 2 : 256 réseaux (8 bits)/64

- Stratégie B : en quatre niveaux égaux :
 - Niveau 1 : 16 réseaux (4 bits) /52
 - Niveau 2 : 16 réseaux (4 bits) /56
 - Niveau 3 : 16 réseaux (4 bits) /60
 - Niveau 4 : 16 réseaux (4 bits) /64

Exemple : Stratégie A en deux niveaux égaux

- fdd4:478f :0611::/48 fournit 256 réseaux fdd4:478f :0611:[0-f][0-f]00:/56 contenant eux-mêmes chacun 256 sous réseaux :
 - fdd4:478f :0611:00[0-f][0-f]:/64 ,
 - fdd4:478f :0611:01[0-f][0-f]:/64 ,
 - fdd4:478f :0611:02[0-f][0-f]:/64 ,
 - fdd4:478f :0611:03[0-f][0-f]:/64 ,
 - ...
 - fdd4:478f :0611:fd[0-f][0-f]:/64 ,
 - fdd4:478f :0611:fe[0-f][0-f]:/64 ,
 - fdd4:478f :0611:ff[0-f][0-f]:/64

Exemple : Stratégie B en quatre niveaux égaux

- fdd4:478f :0611::/48 fournit 16 réseaux fdd4:478f :0611:[0-f]000:/52
 - Contenant eux-mêmes chacun 16 sous réseaux fdd4:478f :0611:[0-f][0-f]00:/56
 - * Contenant eux-mêmes chacun 16 sous réseaux fdd4:478f :0611:[0-f][0-f][0-f]0:/52
 - Contenant eux-mêmes chacun 16 sous réseaux fdd4:478f :0611:[0-f][0-f][0-f][0-f]:/64 ,

9. Plan d'adressage à plusieurs niveaux

En trois niveaux :

- Niveau 1 : 16 (4 bits) /52
- Niveau 2 : 256 (8 bits) /60
- Niveau 3 : 16 (4 bits) /64

ou encore :

- Niveau 1 : 256 (8 bits) /52
- Niveau 2 : 16 (4 bits) /56
- Niveau 3 : 16 (4 bits) /60

On trouvera un exemple de plan d'adressage dans le document IPv6 Subnetting - Overview and Case Study : <https://supportforums.cisco.com/docs/DOC-17232>

10. Plan d'adressage en niveaux inégaux

On vous livre un bloc `fdd4:478f::0611::/48`

Vous disposez d'un site de 4 bâtiments disposant de maximum 8 étages avec plusieurs dizaines de VLANs (/64) par étage. 4 bits pour les bâtiments et 4 bits pour les étages :

- Bâtiment 1 : `0x0000 - 0x0FFF/52`
 - Etage 1 : `0x0000 - 0x00FF/56`
 - Etage 2 : `0x0100 - 0x01FF/56`
 - Etage 3 : `0x0200 - 0x02FF/56`
- Bâtiment 2 : `0x1000 - 0x1FFF/52`
 - Etage 1 : `0x1000 - 0x10FF/56`
 - Etage 2 : `0x1100 - 0x11FF/56`
 - Etage 3 : `0x1200 - 0x12FF/56`
- ...

11. Attribution étalée

“Sparse” mode : attribution étalée

Comme on compte de manière séquentielle par la droite soit `0000`, `0001`, `0010`, `0011`, ... ou `0`, `1`, `2`, `3`, on peut compter par la gauche soit `0000`, `1000`, `0100`, `1100`,... ou `0`, `8`, `4`, `c`, ... par exemple sur le premier hexa pour `fdd4:478f::0611:[0-f]000:/52` :

- `fdd4:478f::0611::/52`
- `fdd4:478f::0611:8000:/52`
- `fdd4:478f::0611:4000:/52`
- `fdd4:478f::0611:c000:/52`
- `fdd4:478f::0611:2000:/52`

Plan d'adressage évolutif réel : <http://www.internetsociety.org/deploy360/resources/ipv6-address-planning-guide-lines-for-ipv6-address-allocation/>

Applications	1er Hexa	Regions	2e Hexa	Unité d'organisation	3e Hexa	Sites	4e Hexa
Data	0	Est	0	Direction	0	Site 1	a
Voice	8	Nord	8	Marketing	1	Site 2	9
Video	4	Ouest	4	Finance	2	Site 3	e
Wireless	c	Sud	c	IT	3	-	-
Management	2	-	-	Support commercial	4	-	-

12. Considérations pratiques sur les adresses IPv6

Une connectivité de type entreprise reçoit un bloc /48 ou /56, par exemple `2001:db8:1ab::/48` ou `2001:db8:1ab::cd00::/56`.

On divise et organise un bloc fixe /48 en 65536 réseaux /64 ou un bloc /56 en 256 réseaux /64.

Les 64 premiers bits, les 3 ou 4 premiers mots sont ceux du réseau (des sous-réseaux) de l'entreprise, qui ne changent

jamais.

Toutes les adresses Unicast doivent avoir un masque de 64 bits. Utiliser un autre masque peut rompre le bon fonctionnement de Neighbor Discovery, Secure Neighbor Discovery (SEND), les Privacy Extensions, Mobile IPv6, Embedded-RP (multicast).

On manipule des données de 16 bits : “word” ou “mot”, parfois “seizet” voire même “doublet” et pourquoi pas le “twoctet” (Wikipedia FR).

Un service DNS dynamique combiné à DHCPv6 sera utile dans un LAN contrôlé.

Les adresses des routeurs et des serveurs doivent être fixes et peuvent être simplifiées, par exemple fe80::88 ou 2001:0db8:00d6:3000::88.

La politique de filtrage du trafic et de sécurité est un enjeu dans un déploiement d'IPv6.

Le dual-stack (la double pile IPv4/IPv6) est toujours la méthode de transition préférée.

En conclusion, on constate en IPv6 :

- Un fonctionnement similaire à IPv4 (préfixe/hôtes).
- Une question de changement culturel :
 - On manipule au maximum 16 bits en hexas : *c'est mieux que 32 bits en binaire.*
 - On peut dépenser des adresses tant qu'on veut : *il n'est plus question de manque ou de restriction.*
 - On utilise les masques de manière extensive (fixée à maximum /64 sur les réseaux) : *il ne faut plus manipuler des masques restrictifs et illisibles.*
- Il s'agit d'un changement inéluctable : *il n'y a pas de protocole alternatif qui autorise la croissance de l'Internet.*

Révisions

Architecture

- Spine-leaf, ACI
- On-premises et cloud, Cloud